

A WENO-based Method of Line Transpose Approach for Vlasov Simulations

Andrew Christlieb¹, Wei Guo², Yan Jiang³

Abstract.

In this paper, a high order implicit Method of Line Transpose (MOL^T) method based on a weighted essentially non-oscillatory (WENO) methodology is developed for one-dimensional linear transport equations and further applied to the Vlasov-Poisson (VP) simulations via dimensional splitting. In the MOL^T framework, the time variable is first discretized by a diagonally implicit strong-stability-preserving Runge-Kutta method, resulting in a boundary value problem (BVP) at the discrete time levels. Then an integral formulation coupled with a high order WENO methodology is employed to solve the BVP. As a result, the proposed scheme is high order accurate in both space and time and free of oscillations even though the solution is discontinuous or has sharp gradients. Moreover, the scheme is able to take larger time step evolution compared with an explicit MOL WENO scheme with the same order of accuracy. The desired positivity-preserving (PP) property of the scheme is further attained by incorporating a newly proposed high order PP limiter. We perform numerical experiments on several benchmarks including linear advection, solid body rotation problem; and on the Landau damping, two-stream instabilities, bump-on-tail, and plasma sheath by solving the VP system. The efficacy and efficiency of the proposed scheme is numerically verified.

Key Words: Method of Line Transpose, implicit Runge-Kutta methods, weighted essentially non-oscillatory methodology, high order accuracy, Vlasov-Poisson, dimensional splitting, positivity-preserving

¹ Department of Mathematics and Electrical Engineering, Michigan State University, East Lansing, MI, 48824. E-mail: andrewch@math.msu.edu

² Department of Mathematics, Michigan State University, East Lansing, MI, 48824. E-mail: wguo@math.msu.edu

³ Department of Mathematics, Michigan State University, East Lansing, MI, 48824. E-mail: jiangyan@math.msu.edu

1 Introduction

Transport modeling plays an important role in a wide range of applications, including fluid dynamics, plasma physics, atmospheric sciences, among many others. The main focus of this paper is to develop of efficient high order transport solvers for the Vlasov simulations. As a high-dimensional kinetic transport equation, the Vlasov-Poisson (VP) system is considered a fundamental model in plasma physics which describes the dynamics of charged particles due to the self-consistent electric force:

$$f_t + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E}(\mathbf{x}, t) \cdot \nabla_{\mathbf{v}} f = 0, \quad \mathbf{x} \times \mathbf{v} \in \Omega_{\mathbf{x}} \times \Omega_{\mathbf{v}} \quad (1.1a)$$

$$\mathbf{E}(\mathbf{x}, t) = -\nabla_{\mathbf{x}} \phi(\mathbf{x}, t), \quad -\Delta_{\mathbf{x}} \phi(\mathbf{x}, t) = -1 + \rho(\mathbf{x}, t) \quad (1.1b)$$

where $f(\mathbf{x}, \mathbf{v}, t)$ describes the probability of finding a particle with velocity \mathbf{v} at position \mathbf{x} at time t , \mathbf{E} is the electrostatic field, ϕ is the self-consistent electrostatic potential, and $\rho(\mathbf{x}, t) = \int_{\Omega_{\mathbf{v}}} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}$ is the electron charge density and the 1 represents the uniformly distributed infinitely massive ions in the background.

The VP simulation is challenging. Besides the famous curse of dimensionality, the underlying system may develop filamentation solution structures, since the long-range forces drive the solution out of thermo-equilibrium. Hence, it is of importance that the used numerical scheme is able to effectively capture filamentation structures without producing spurious oscillations. Further, the VP system conserves several physical quantities including the total mass, momentum, and energy, which requires that the numerical scheme conserves those invariants on the discrete level. In addition, as a probability density function in phase space, the unknown function f always remains non-negative. In practice, the positivity preserving property is another highly desired property for the design of Vlasov solver, since negative solutions may trigger unphysical phenomenon in plasmas.

There exist a large amount of successful VP solvers in the literature, such as the prominent particle-in-cell (PIC) method [4, 23, 41], which evolves the solution by following the trajectories of some sampled macro-particles, Eulerian type approaches [18, 50, 12, 11], for which the state variable is advanced according to the underlying equations on a fixed numerical grid, and semi-Lagrangian approach [9, 29, 2, 39, 19, 3, 17, 16, 34, 30, 15, 22], which updates the solution by tracking the characteristics on a fixed mesh grid. However, these popular schemes still have limitations. For example, the PIC method is known to suffer from low order sampling noise; Eulerian approaches are

subject to stringent time step restriction associated with explicit time stepping; general boundary conditions are difficult to handle in a semi-Lagrangian setting. Also note that many of these existing methods are in the Method of Line (MOL) framework, meaning that the spatial variable is first discretized, then the numerical solution is updated in time by coupling a suitable time integrator. In this paper, we consider an alternative approach to advance the solution, which is called the Method of Line Transpose (MOL^T) or Rothes's method in the literature [35, 36]. As its name implies, such an approach is defined in an orthogonal fashion of the MOL schemes, i.e., the discretization is first carried out for temporal variable, resulting in a boundary value problem (BVP) at the discrete time levels. Then a preferred BVP solver can be further applied to advance the solution. For example, in [6], an implicit integral formulation based on the MOL^T framework for one-dimensional wave equations is developed and analyzed. It is proven that the scheme is A-stable, hence allowing for extra large time step evolution. In [8], the computational cost of the scheme is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, where N is the number of discrete mesh points, by taking advantage of an important property of the analytical solution of the wave equations. More recently, a successive convolution technique [5] is developed to enhance the temporal accuracy to arbitrary high order. A notable advantage of such a scheme is that, even though it is implicit in time, we do not need to explicitly solve a linear system, while the BVP is inverted analytically in an integral formulation. The method is further extended to multi-dimensional wave equations by utilizing the alternating direction implicit strategy at the expense of introducing splitting errors [6]. However, the simplicity of the scheme is preserved. The recent applications of this MOL^T approach include the heat equation, the Allen-Cahn equation [7], and Maxwell's equations [10].

In this paper, we consider formulating a numerical scheme suited for the Vlasov equation in the MOL^T framework in order to take advantage of its good properties such as the simplicity and the ability to take large time steps for evolution. In order to accommodate the MOL^T approach, the nonlinear multi-dimensional Vlasov equation is first split into several lower-dimensional linear transport equations. Such an idea has been widely used in semi-Lagrangian Vlasov simulations [9]. As we mentioned earlier, the extra difficulty of VP simulations lies in the distinctive filamentation solution structures, which requires that the employed numerical scheme is able to effectively capture sharp gradient of the solution without producing spurious oscillations even on an under-resolved mesh. Note that we cannot directly apply the existing MOL^T schemes to the VP system, since

they are all linear schemes and hence exhibit oscillations when the solution is discontinuous or has a sharp gradient. In order to control undesired oscillations, we propose to further couple two numerical ingredients. Firstly, in the MOL^T framework, the time variable is discretized via an implicit high order strong-stability-preserving (SSP) Runge-Kutta (RK) method [21]. Secondly, a robust weighted essential non-oscillatory (WENO) methodology is incorporated when inverting the BVP. We expect that the resulting scheme is efficient, high order accurate, robust, and able to take large time step evolution (larger than the explicit counterpart). Furthermore, a high order positivity-preserving (PP) limiter suited for the proposed MOL^T scheme is developed to ensure such a high desired property for the numerical solution.

We also would like to point out that the proposed MOL^T WENO scheme shares some similarity with the semi-Lagrangian WENO scheme developed in [31, 32], in the sense that both schemes are high order accurate and able to effectively capture the filamentation structures with large time step. On the other hand, compared with semi-Lagrangian WENO schemes, the proposed method is able to conveniently handle general boundary conditions, which is not trivial for semi-Lagrangian approaches.

The rest of the paper is organized as follows. In Section 2, we formulate the numerical scheme for one-dimensional linear transport equations by coupling several main numerical ingredients such as the MOL^T approach, the WENO methodology and the high order PP limiter. In Section 3, the method is extended to two-dimensional transport equations via dimensional splitting. We present several benchmark tests in Section 4 to verify the performance of the proposed scheme, including linear transport, rigid body rotation, as well as Landau damping, two-stream instabilities, bump-on-tail and plasma sheath for VP simulations. We conclude the paper in Section 5.

2 Formulation of the Scheme

We start with the following one-dimensional advection equation

$$u_t + cu_x = 0, \quad x \in [a, b], \quad (2.2)$$

where c is constant and denotes the wave propagation speed. The boundary condition is assumed to be periodic, Dirichlet boundary condition

$$u(a, t) = g_1(t), \text{ for } c > 0, \quad \text{or} \quad u(b, t) = g_2(t), \text{ for } c < 0,$$

or Neumann boundary condition

$$u_x(a, t) = h_1(t), \text{ for } c > 0, \quad \text{or} \quad u_x(b, t) = h_2(t), \text{ for } c < 0.$$

2.1 The method of line transpose approach

As discussed in the introduction, in a MOL^T framework, we first discretize the time variable while leaving the spatial variable continuous. For example, we can apply the first order backward Euler method to (2.2), and obtain

$$\frac{u^{n+1} - u^n}{\Delta t} + cu_x^{n+1} = 0,$$

i.e.,

$$u_x^{n+1} + \text{sgn}(c) \cdot \alpha u^{n+1} = \text{sgn}(c) \cdot \alpha u^n, \quad (2.3)$$

where Δt denotes the time step and $\alpha = \frac{1}{|c\Delta t|}$. Note that (2.3) is an initial value problem (IVP), for which we can obtain an explicit representation for solution u^{n+1} at time level t^{n+1} . If we assume $c > 0$, u^{n+1} can be represented by

$$u^{n+1}(x) = I^L[u^n, \alpha](x) + A^{n+1}e^{-\alpha(x-a)}, \quad (2.4)$$

where

$$I^L[v, \alpha](x) \doteq \alpha \int_a^x e^{-\alpha(x-y)} v(y) dy, \quad (2.5)$$

and A^{n+1} is a constant and should be determined by the boundary condition. The superscript L indicates that the characteristics traverse from the left to the right. If $c < 0$, then the characteristics traverse from the right to the left. Due to the symmetry, we can advance the solution as

$$u^{n+1}(x) = I^R[u^n, \alpha](x) + B^{n+1}e^{-\alpha(b-x)}, \quad (2.6)$$

when the first order backward Euler scheme is used. Here,

$$I^R[v, \alpha](x) \doteq \alpha \int_x^b e^{-\alpha(y-x)} v(y) dy, \quad (2.7)$$

and again the constant B^{n+1} is determined by the boundary condition.

We need to further discretize the spatial variable based on the semi-discrete schemes (2.4) and (2.6). Following the idea in [6, 8], the domain $[a, b]$ is discretized by $M + 1$ uniformly distributed grid points

$$a = x_0 < x_1 < \cdots < x_{M-1} < x_M = b,$$

and $\Delta x = \frac{b-a}{M}$ denotes the mesh size. The solution is accordingly discretized at points $\{x_i\}_{i=0}^M$, and we use u_i^n to denote the numerical solution at location x_i at time level t^n . The main part of the algorithm is to evaluate the convolution integral $I^L[v, \alpha](x)$ in (2.4), (2.6) or $I^R[v, \alpha]$ in (2.5), (2.7) based on the point values $\{v_i\}_{i=0}^M$.

We first consider the case of $c > 0$. Note that,

$$I^L[v, \alpha](x) = I^L[v, \alpha](x - \delta_x)e^{-\alpha\delta_x} + J^L[v, \alpha, \delta_x](x), \quad (2.8)$$

where

$$J^L[v, \alpha, \delta_x](x) \doteq \alpha \int_{x-\delta_x}^x v(y)e^{-\alpha(x-y)} dy, \quad (2.9)$$

and δ_x is a positive constant. Therefore, in order to obtain $I^L[v, \alpha](x_i)$ denoted by I_i^L , we only need to approximate the increment $J^L[v, \alpha, \Delta x](x_i)$ denoted by J_i^L , and obtain I_i^L via a recursive relation, i.e.,

$$I_i^L = I_{i-1}^L e^{-\alpha\Delta x} + J_i^L, \quad i = 1, \dots, M, \quad I_0^L = 0. \quad (2.10)$$

For the case of $c < 0$, again thanks to the symmetry, we have the following identity

$$I^R[v, \alpha](x) = I^R[v, \alpha](x + \delta_x)e^{-\alpha\delta_x} + J^R[v, \alpha, \delta_x](x), \quad (2.11)$$

where

$$J^R[v, \alpha, \delta_x](x) \doteq \alpha \int_x^{x+\delta_x} v(y)e^{-\alpha(y-x)} dy, \quad (2.12)$$

and hence a similar recursive relation can be obtained

$$I_i^R = I_{i+1}^R e^{-\alpha\Delta x} + J_i^R, \quad i = 0, \dots, M-1, \quad I_M^R = 0, \quad (2.13)$$

where I_i^R and J_i^R denote the approximate values of $I^R[v, \alpha](x_i)$ and $J^R[v, \alpha, \Delta x](x_i)$, respectively.

After computing the contributions from the left or right characteristic, we further need to find the global coefficient A in (2.4) or B in (2.6), which is used to enforce the boundary condition. Since $I^L[u^n, \alpha](a) = I^R[u^n, \alpha](b) = 0$, for a Dirichlet boundary condition, the global coefficient can be obtained by

$$A^{n+1} = g_1(t^{n+1}), \text{ or } B^{n+1} = g_2(t^{n+1}). \quad (2.14)$$

Similarly, for a Neumann boundary condition, because $(I^L[v, \alpha])'(x) = -\alpha I^L[v, \alpha](x) + \alpha v(x)$ and $(I^R[v, \alpha])'(x) = \alpha I^R[v, \alpha](x) - \alpha v(x)$, we can get the coefficients by

$$A^{n+1} = v(a) - \frac{1}{\alpha} h_1(t^{n+1}), \text{ or } B^{n+1} = v(b) + \frac{1}{\alpha} h_2(t^{n+1}). \quad (2.15)$$

For the periodic boundary condition, the coefficient can be determined by using the mass conservation property of the solution, i.e.,

$$\int_a^b u(x, t) dx = \int_a^b u(x, 0) dx.$$

On the discrete level, we enforce

$$\sum_{i=0}^{M-1} u_i^n = \sum_{i=0}^{M-1} u_i^{n+1} = \sum_{i=0}^{M-1} I_i^L + A^{n+1} \sum_{i=0}^{M-1} e^{-i\alpha\Delta x}, \quad u_M^{n+1} = u_0^{n+1}, \quad \text{for } c > 0,$$

or

$$\sum_{i=1}^M u_i^n = \sum_{i=1}^M u_i^{n+1} = \sum_{i=1}^M I_i^R + B^{n+1} \sum_{i=1}^M e^{-(M-i)\alpha\Delta x}, \quad u_0^{n+1} = u_M^{n+1}, \quad \text{for } c < 0.$$

Then the coefficient is obtained

$$A^{n+1} = \left(\sum_{i=0}^{M-1} u_i^n - \sum_{i=0}^{M-1} I_i^L \right) / \sum_{i=0}^{M-1} e^{-i\alpha\Delta x}, \quad (2.16)$$

or

$$B^{n+1} = \left(\sum_{i=1}^M u_i^n - \sum_{i=1}^M I_i^R \right) / \sum_{i=1}^M e^{-(M-i)\alpha\Delta x}. \quad (2.17)$$

2.2 WENO methodology

The existing MOL^T schemes mainly use linear interpolation methods to compute increment J_i^L and J_i^R such as [6], which work well for smooth problems. However, the schemes may generate spurious oscillations when the solution has discontinuities because of the famous Gibbs phenomenon. In order to control such undesired oscillations, we will incorporate a WENO methodology in the MOL^T framework to evaluate J_i^L and J_i^R below.

The first WENO scheme was introduced by Liu *et al.* [27], as a third order accurate finite volume scheme to solve hyperbolic conservation laws in one dimension. Later, in [24], a general framework for arbitrary order accurate finite difference WENO schemes was designed, which is more efficient for multi-dimensional calculations. The schemes have the advantage of attaining high order accuracy in smooth regions of the solution while maintaining sharp and essentially non-oscillatory transitions of discontinuities. More information about WENO can be found in [38].

In our work, a WENO integration procedure is used, based on the idea proposed in [13, 14, 28]. Below, a $(2k - 1)$ -th order approximation of J_i^L is provided as an example, where k is an integer. The corresponding coefficients of the third order and fifth order schemes are given in the Appendix. The whole procedure consists of four steps.

1. On each small stencil $S_r(i) = \{x_{i-r-1}, \dots, x_{i-r-1+k}\}$, $r = 0, \dots, k-1$, which contains x_{i-1} and x_i , there is a unique polynomial $p_r(x)$ of degree at most k which interpolates $v(x)$ at the nodes in $S_r(i)$. Then we can get the integral

$$J_{i,r}^L = \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-y)} p_r(y) dy = \sum_{j=0}^k c_{i-r-1+j}^{(r)} v_{i-r-1+j}, \quad (2.18)$$

where the coefficients $c_{i-r-1+j}^{(r)}$ depend on α and the cell size Δx , but not on v .

2. Similarly, on the big stencil $S(i) = \{x_{i-k}, \dots, x_{i+k-1}\}$, there is a unique polynomial $p(x)$ of degree at most $2k-1$ interpolating $v(x)$ at the nodes in $S(i)$. Then we have

$$J_i^L = \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-y)} p(y) dy = \sum_{j=0}^{2k-1} c_{i-k+j} v_{i-k+j}, \quad (2.19)$$

and linear weights d_r , such that

$$J_i^L = \sum_{r=0}^{k-1} d_r J_{i,r}^L, \quad (2.20)$$

where $\sum_{r=0}^{k-1} d_r = 1$.

3. Change the linear weights d_r into nonlinear weights ω_r , using

$$\omega_r = \frac{\tilde{\omega}_r}{\sum_{s=0}^{k-1} \tilde{\omega}_s}, \quad r = 0, \dots, k-1, \quad (2.21)$$

with

$$\tilde{\omega}_r = \frac{d_r}{(\epsilon + \beta_r)^2}.$$

Here, $\epsilon > 0$ is a small number to avoid the denominator becoming zero, and we take $\epsilon = 10^{-6}$ in our numerical tests. The smoothness indicator β_r , measuring the relative smoothness of the function $v(x)$ in the stencil $S_r(i)$, is defined by

$$\beta_r = \sum_{l=1}^k \int_{x_{i-1}}^{x_i} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2 dx, \quad (2.22)$$

4. Lastly, we have

$$J_i^L = \sum_{r=0}^{k-1} \omega_r J_{i,r}^L. \quad (2.23)$$

The process to obtain J_i^R is mirror-symmetric to that for J_i^L , with respect to the point x_i .

When dealing with a Dirichlet or a Neumann boundary condition, we further need to extrapolate point values at the ghost points $\{x_{-(k-1)}, \dots, x_{-1}\}$ and $\{x_{M+1}, \dots, x_{M+k-1}\}$ using the point values in the domain, where

$$x_{-i} = x_0 - i\Delta x, \quad x_{M+i} = x_M + i\Delta x.$$

Following the idea in [40], a WENO extrapolation strategy is designed to get more robust results. For instance, when approximating the left ghost point x_{-i} , we consider the big stencil $S_{2k-2} = \{x_0, \dots, x_{2k-2}\}$:

1. On each small stencil $S_r = \{x_0, \dots, x_r\}$, $r = 0, \dots, 2k-2$, we can construct a polynomial $p_r(x)$ of degree at most r interpolating $v(x)$ at the nodes in S_r . And v_{-i} can be extrapolated by

$$v_{-i} = \sum_{r=0}^{2k-2} d_r p_r(x_{-i}), \quad (2.24)$$

where $d_0 = \Delta x^{2k-2}$, $d_1 = \Delta x^{2k-3}$, \dots , $d_{2k-3} = \Delta x$, and $d_{2k-2} = 1 - \sum_{r=0}^{2k-3} d_r$.

2. Change the weights d_r into nonlinear weights ω_r

$$\omega_r = \frac{\tilde{\omega}_r}{\sum_{s=0}^{2k-2} \tilde{\omega}_s}, \quad r = 0, \dots, 2k-2, \quad (2.25)$$

with

$$\tilde{\omega}_r = \frac{d_r}{(\epsilon + \beta_r)^2}.$$

Again, we take $\epsilon = 10^{-6}$ in our numerical tests to avoid the denominator becoming zero. And the smoothness indicator β_r is defined by

$$\begin{aligned} \beta_0 &= \Delta x^2, \\ \beta_r &= \sum_{l=1}^r \int_{x_{-1}}^{x_0} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2 dx, \quad r = 1, \dots, 2k-2. \end{aligned}$$

3. Finally, we have

$$v_{-i} = \sum_{r=0}^{2k-2} \omega_r p_r(x_{-i}).$$

Similar, we can get the WENO extrapolation on right ghost points x_{M+i} based on $S_{2k-2} = \{x_{M-2k+2}, \dots, x_M\}$.

2.3 High order time discretization

Note that scheme (2.3) is first order accurate in time. Below, we will extend our schemes to high order accuracy via the implicit SSP RK method [21, 25].

To approximate the solutions of ordinary differential equation

$$u_t = F(u), \quad (2.26)$$

an s -stage and k -th order RK method, denoted as $RK(s, k)$, is defined as

$$u^{(i)} = u^n + \Delta t \sum_{j=1}^s a_{ij} F(u^{(j)}, t_n + c_j \Delta t), \quad i \leq j \leq s, \quad (2.27a)$$

$$u^{n+1} = u^n + \Delta t \sum_{j=1}^s b_j F(u^{(j)}, t_n + c_j \Delta t), \quad (2.27b)$$

with the standard assumption

$$c_i = \sum_{j=1}^s a_{ij}.$$

Denote the matrix $\mathbf{A} = (a_{ij})_{s \times s}$ and the vector $\mathbf{b} = (b_i)_{s \times 1}$. In our scheme, we will use the diagonally implicit (DIRK) methods, for which \mathbf{A} is lower triangular. Denote $\mathbf{1} = [1, \dots, 1]^T \in R^s$, $\mathbf{U}^{rk} = [u^{(1)}, \dots, u^{(s)}]^T$, $\mathbf{F} = [\Delta t F(u^{(1)}, t_n + c_1 \Delta t), \dots, \Delta t F(u^{(s)}, t_n + c_s \Delta t)]^T$, and $\Lambda = \text{diag}(a_{11}, \dots, a_{ss})$. Then, we can get $\mathbf{F} = \mathbf{A}^{-1}(\mathbf{U}^{rk} - u^n \cdot \mathbf{1})$. Hence

$$\mathbf{U}^{rk} - \Lambda \cdot \mathbf{F} = u^n \cdot \Lambda \cdot \mathbf{A}^{-1} \cdot \mathbf{1} + (\mathbf{A} - \Lambda) \cdot \mathbf{A}^{-1} \mathbf{U}^{rk} \quad (2.28a)$$

$$u^{n+1} = u^n + \mathbf{b}^T \cdot \mathbf{F} = u^n + \mathbf{b}^T \cdot \mathbf{A}^{-1}(\mathbf{U}^{rk} - u^n \cdot \mathbf{1}) \quad (2.28b)$$

Note that \mathbf{A} is lower triangular. So $(\mathbf{A} - \Lambda) \cdot \mathbf{A}^{-1}$ is also lower triangular, with all diagonal elements equal to zero.

For problem (2.2), $F(u) = -cu_x$. Corresponding, each stage of (2.28a) can be rewritten as

$$u_x^{(i)} + \text{sgn}(c \cdot a_{ii}) \alpha_i u^{(i)} = \tilde{\alpha}_i u^n + \sum_{j=1}^{i-1} \tilde{\alpha}_{ij} u^{(j)}, \quad (2.29)$$

where $\alpha_i = 1/|c \cdot a_{ii} \Delta t|$. We can solve (2.29) with MOL^T method mentioned above. And finally, u^{n+1} is obtained by (2.28b).

When problem (2.2) combining with the periodic boundary condition, the global coefficients $A^{rk,i}$ or $B^{rk,i}$ on each stage can be obtained by (2.16) or (2.17). For a Dirichlet boundary condition,

the scheme may lose accuracy if we define $A^{rk,i} = g_1(t_{rk}^i)$ or $B^{rk,i} = g_2(t_{rk}^i)$. To avoid order reduction, we follow the idea in [1]. In particular, denote $\mathbf{c}^l = [c_1^l, \dots, c_s^l]^T$ for $l > 0$. To achieve k -th order of accuracy, boundary values at intermediate stages are defined as

$$\mathbf{A}^{rk} = \sum_{l=0}^{k-2} (\mathbf{A}^l \mathbf{1} \otimes \Delta t^l \mathbf{c}^l) \frac{d^l}{dt^l} g_1(t_n) + (\mathbf{A}^{k-1} \otimes \Delta t^{k-1} \mathbf{c}^{k-1}) \frac{d^{k-1}}{dt^{k-1}} \mathbf{G}_1^{n+c} \quad (2.30)$$

for $c > 0$, where

$$\begin{aligned} \mathbf{A}^{rk} &= [A^{rk,1}, \dots, A^{rk,s}]^T, \\ \mathbf{G}_1^{n+c} &= [g_1(t^n + c_1 \Delta t), \dots, g_1(t^n + c_s \Delta t)]^T, \end{aligned}$$

or

$$\mathbf{B}^{rk} = \sum_{l=0}^{k-2} (\mathbf{A}^l \mathbf{1} \otimes \Delta t^l \mathbf{c}^l) \frac{d^l}{dt^l} g_2(t_n) + (\mathbf{A}^{k-1} \otimes \Delta t^{k-1} \mathbf{c}^{k-1}) \frac{d^{k-1}}{dt^{k-1}} \mathbf{G}_2^{n+c} \quad (2.31)$$

for $c < 0$, where

$$\begin{aligned} \mathbf{B}^{rk} &= [B^{rk,1}, \dots, B^{rk,s}]^T, \\ \mathbf{G}_2^{n+c} &= [g_2(t^n + c_1 \Delta t), \dots, g_2(t^n + c_s \Delta t)]^T. \end{aligned}$$

The similar idea is used when dealing with a Neumann boundary condition.

In our numerical tests, we will use $RK(2, 3)$ and $RK(4, 4)$ schemes to obtain third order accuracy and fourth order accuracy in time, respectively. Corresponding coefficients are listed as appendix.

2.4 Positivity-preserving limiter

The exact solution of (2.2) satisfies the positivity preserving principle, meaning that if the initial condition $u(x, 0) \geq 0$, then the exact solution $u(x, t) \geq 0$ for any time $t > 0$. In practice, it is highly desirable to have numerical schemes also satisfy such a property on the discrete level. It is even more crucial for the Vlasov simulation, since negative values may trigger some nonphysical process in plasmas [33]. Recently, Zhang *et al.* developed a high order positivity-preserving limiters for finite volume WENO methods and finite element DG methods, which is able to maintain high order accuracy of the original scheme and easy to implement [46, 47, 48, 49]. Later, Xu *et al.* [45, 26, 42, 43, 44] developed a parameter maximum principle preserving flux limiter for finite difference WENO schemes by limiting the high order flux towards a first order monotone flux.

However, both limiters are exclusively designed for MOL type methods and hence cannot be directly applied in the MOL^T setting. Below, we will formulate a new positivity-preserving (PP) limiter suited for the proposed WENO MOL^T scheme, and also show that such a limiter is able to preserve positive while maintaining the original high order accuracy.

Assume that we have $u_i^n \geq 0$ for all i at time t^n . After obtaining the numerical solutions u_i^{n+1} , we can rewrite our scheme as a conservative form

$$u_i^{n+1} = u_i^n - \Delta t / \Delta x (\hat{f}_{i+1/2} - \hat{f}_{i-1/2}),$$

in which the artificial numerical fluxes $\hat{f}_{i+1/2}$ are recursively defined by

$$\hat{f}_{i+1/2} = \begin{cases} 0, & i = -1, \\ \hat{f}_{i-1/2} - \Delta x / \Delta t (u_i^{n+1} - u_i^n), & i = 0, \dots, M. \end{cases}$$

If we are able to modify the numerical flux $\hat{f}_{i+1/2}$ into $\tilde{f}_{i+1/2}$, such that

$$u_i^{new} = u_i^n - \Delta t / \Delta x (\tilde{f}_{i+1/2} - \tilde{f}_{i-1/2}) \geq 0, \quad i = 0, \dots, M, \quad (2.32)$$

then the modified solution u^{new} at time level $t = t^{n+1}$ will be non-negative automatically. Roughly speaking, the main idea of the limiter is to enforce the negative solution value back to zero by decreasing the outflow flux. For instance, when $c > 0$, we will reset the numerical flux $\hat{f}_{i+1/2}$ if $u_i^{n+1} < 0$. Details of the limiter are shown below.

Firstly, we consider the case $c > 0$ and a Dirichlet boundary condition. Numerical flux $\hat{f}_{i+1/2}$ is modified into $\tilde{f}_{i+1/2}$, starting from $i = 1$ to $i = M$,

$$\tilde{f}_{i+1/2} = \begin{cases} \tilde{f}_{i-1/2} + \Delta x / \Delta t u_i^n, & \text{if } u_i^{n+1,1} < 0, \\ \hat{f}_{i+1/2}, & \text{otherwise,} \end{cases} \quad (2.33)$$

where

$$u_i^{n+1,1} = u_i^n - \Delta t / \Delta x (\hat{f}_{i+1/2} - \tilde{f}_{i-1/2}).$$

$\tilde{f}_{-1/2} = \hat{f}_{-1/2}$ and $\tilde{f}_{1/2} = \hat{f}_{1/2}$, due to the Dirichlet boundary condition. For a Nuemann boundary condition, we can also use (2.33) to modify fluxes. Once the modified fluxes are obtained, the positivity-preserving solution is computed by (2.32).

For periodic boundary conditions, however, we need an additional step in order to preserve the positivity and periodicity of the solution. In particular, we first modify the numerical flux $\hat{f}_{i+1/2}$ into $\hat{\hat{f}}_{i+1/2}$ by (2.33) i.e., starting from $i = 0$ into $i = M - 1$, with

$$\hat{\hat{f}}_{i+1/2} = \begin{cases} \hat{f}_{i-1/2} + \Delta x / \Delta t u_i^n, & \text{if } u_i^{n+1,1} < 0, \\ \hat{f}_{i+1/2}, & \text{otherwise,} \end{cases}$$

where

$$u_i^{n+1,1} = u_i^n - \Delta t / \Delta x (\hat{f}_{i+1/2} - \hat{\tilde{f}}_{i-1/2}).$$

For periodicity, we further set $\tilde{f}_{-1/2} = \hat{\tilde{f}}_{M-1/2}$. Note that since the flux at $x_{-1/2}$ is modified, the solution at x_0 may become negative. Hence, we need screen the solution one more time by the following procedure. Starting $i = 0$, we compute

$$u_i^{n+1,2} = u_i^n - \Delta t / \Delta x (\hat{\tilde{f}}_{i+1/2} - \tilde{f}_{i-1/2}).$$

If $u_i^{n+1,2} < 0$, we let

$$\tilde{f}_{i+1/2} = \tilde{f}_{i-1/2} + \Delta x / \Delta t u_i^n.$$

Otherwise, let $i_0 = i$ and for any j with $i_0 \leq j \leq M - 1$, let $\tilde{f}_{j+1/2} = \hat{\tilde{f}}_{j+1/2}$. Finally, we are able to obtain the positivity-preserving solution u^{new} via (2.32). Note that i_0 must be less than $M - 1$ and hence $\tilde{f}_{M-1/2} = \tilde{f}_{-1/2}$.

When $c < 0$, wind blows from right to left, and we will reset the numerical flux $\hat{f}_{i-1/2}$ if $u_i^{n+1} < 0$. Similar idea can be used for different boundary conditions.

Below, we show that the PP-limiter can preserve positivity while maintaining the previous high order. In particular, we have the following proposition.

Proposition 2.1. Consider the WENO MOL^T scheme is used for solving (2.2) and assume that the numerical solution u_i^{n+1} is p -th order accurate in space and q -th order accurate in time, i.e., the global error

$$e_i^{n+1} = |u_i^{n+1} - u(x_i, t^{n+1})| = \mathcal{O}(\Delta x^p + \Delta t^q), \quad \forall i,$$

where $u(x_i, t^{n+1})$ is the exact solution. If the proposed PP-limiter (2.32) with (2.33) is applied, then

$$\tilde{e}_i^{n+1} = |u_i^{new} - u(x_i, t^{n+1})| = \mathcal{O}(\Delta x^{p-1} + \Delta x^{-1} \Delta t^q), \quad \forall i$$

for the worst case, when $\Delta t = \mathcal{O}(\Delta x)$.

Proof. Without loss of generality, we only consider (2.2) with $c > 0$ and a Dirichlet boundary condition is imposed for brevity. Let u_i^{n+1} be the first solution value that is less than zero starting from u_1^{n+1} , i.e., $u_i^{n+1} < 0$ and $u_j^{n+1} \geq 0$, $1 \leq j < i$. Then we have $\tilde{f}_{j-1/2} = \hat{f}_{j-1/2}$, $0 \leq j \leq i$,

and $\tilde{f}_{i+1/2} = \tilde{f}_{i-1/2} + \Delta x / \Delta t u_i^n$, such that $u_i^{new} = 0$. Recalling that $u(x_i, t^{n+1}) \geq 0$ and $e_i^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q)$, and hence $\tilde{e}_i^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q)$. Also note that the modification of flux

$$\tilde{f}_{i+1/2} - \hat{f}_{i+1/2} = \Delta x / \Delta t u_i^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q)$$

is a negative high order quantity.

Now, we consider the effect of the modification of flux $\tilde{f}_{i+1/2}$ to the next point value in terms of accuracy. First, we need to compute

$$u_{i+1}^{n+1,1} = u_{i+1}^n - \Delta t / \Delta x (\hat{f}_{i+3/2} - \tilde{f}_{i+1/2}).$$

Several cases need to be taken into account.

Case (1) If $u_{i+1}^{n+1,1} \geq 0$. Then, $\tilde{f}_{i+3/2} = \hat{f}_{i+3/2}$ by (2.33), and $u_{i+1}^{new} = u_{i+1}^{n+1,1}$. We have

$$|u_{i+1}^{new} - u_{i+1}^{n+1}| = |\Delta t / \Delta x (\tilde{f}_{i+1/2} - \hat{f}_{i+1/2})| = \mathcal{O}(\Delta x^p + \Delta t^q).$$

Hence,

$$\tilde{e}_{j+1}^{n+1} = |u_{i+1}^{new} - u(x_{i+1}, t^{n+1})| \leq |u_{i+1}^{new} - u_{i+1}^{n+1}| + e_{i+1}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q).$$

Case (2) If $u_{i+1}^{n+1,1} < 0$. Then, $\tilde{f}_{i+3/2} = \tilde{f}_{i+1/2} + \Delta x / \Delta t u_{i+1}^n$ by (2.33), and $u_{i+1}^{new} = 0$. There are two cases we need to treat differently. If $u_{i+1}^{n+1} < 0$, then by the same argument we used to treat the case at x_i , we have $u_{i+1}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q)$ and hence $e_{i+1}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q)$. If $u_{i+1}^{n+1} \geq 0$, then we have

$$|u_{i+1}^{n+1,1} - u_{i+1}^{n+1}| = |\Delta t / \Delta x (\tilde{f}_{i+1/2} - \hat{f}_{i+1/2})| = \mathcal{O}(\Delta x^p + \Delta t^q).$$

Further, since $u_{i+1}^{n+1,1} < 0$ and $u_{i+1}^{n+1} > 0$, combining the above estimate, we have

$$|u_{i+1}^{n+1}| = \mathcal{O}(\Delta x^p + \Delta t^q).$$

Consequently,

$$\tilde{e}_{j+1}^{n+1} = |u_{i+1}^{new} - u(x_{i+1}, t^{n+1})| = |u(x_{i+1}, t^{n+1})| \leq |u_{i+1}^{n+1}| + e_{j+1}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q).$$

We also have the following modification of the flux

$$\tilde{f}_{i+3/2} - \hat{f}_{i+3/2} = \Delta x / \Delta t u_i^{n+1} + \tilde{f}_{i+1/2} - \hat{f}_{i+1/2} = \Delta x / \Delta t (u_{i+1}^{n+1} + u_i^{n+1}) = \mathcal{O}(\Delta x^p + \Delta t^q). \quad (2.34)$$

For both cases, we have shown that $\tilde{e}_{j+1}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q)$, i.e., the PP-limiter preserves the order of accuracy. Moreover, for case (1) since the flux $\hat{f}_{i+3/2}$ is not changed, we can move on to

consider the solution value at x_{i+2} with the same procedure as the case at x_i . The situation of case (2) is more complicated when considering the subsequent effect of the modification of flux (2.34). Again, we need to consider two cases.

Case (2.1) If $u_{i+2}^{n+1,1} \geq 0$. Similar to case (1) above, the flux at $\tilde{f}_{i+5/2}$ should remain the same, i.e., $\tilde{f}_{i+5/2} = \hat{f}_{i+5/2}$. By (2.34)

$$|u_{i+2}^{new} - u_{i+2}^{n+1}| = |\Delta t / \Delta x (\tilde{f}_{i+3/2} - \hat{f}_{i+3/2})| = \mathcal{O}(\Delta x^p + \Delta t^q),$$

and consequently,

$$\tilde{e}_{i+2}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q).$$

Case (2.2) If $u_{i+2}^{n+1,1} < 0$. We can repeat the analysis of case (2) and obtain that

$$|u_{i+2}^{n+1}| = \mathcal{O}(\Delta x^p + \Delta t^q),$$

which gives the estimate

$$\tilde{e}_{i+2}^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q),$$

and the modification of the flux

$$\tilde{f}_{i+5/2} - \hat{f}_{i+5/2} = \Delta x / \Delta t (u_{i+2}^{n+1} + u_{i+1}^{n+1} + u_i^{n+1}) = \mathcal{O}(\Delta x^p + \Delta t^q). \quad (2.35)$$

More generally, assume that the fluxes are successively modified with at most constant many times, i.e., there exists a constant K_0 independent of Δx and Δt , such that if the fluxes $\tilde{f}_{i+k+1/2}$ for $0 \leq k \leq K$ are successively changed based on

$$\tilde{f}_{i+k+1/2} = \hat{f}_{i+k+1/2} + \Delta x / \Delta t \sum_{l=0}^k u_{i+l}^{n+1},$$

then K is always bounded by K_0 . We have

$$\tilde{e}_i^{n+1} = \mathcal{O}(\Delta x^p + \Delta t^q).$$

For the worst cases, the fluxes may be successively modified with $\mathcal{O}(1/\Delta x)$ times, then the accumulation of error leads to one order reduction in space, i.e.,

$$\tilde{e}_i^{n+1} = \mathcal{O}(\Delta x^{p-1} + \Delta x^{-1} \Delta t^q).$$

Remark 2.2. In our simulations, the worst cases never happened: K_0 is always a constant independent of mesh sizes, and optimal accuracy is observed for all our accuracy tests.

3 Two-dimensional implementation

In this section, we extend the algorithm to two-dimensional problems in the following form:

$$u_t + f(y, t)u_x + g(x, t)u_y = 0 \quad (3.36)$$

In particular, (3.36) is solved in a splitting framework. First, we decouple the 2D advection equation into a sequence of independent 1D advection equations

$$u_t + f(y, t)u_x = 0, \quad (3.37a)$$

$$u_t + g(x, t)u_y = 0. \quad (3.37b)$$

Let Q_1 and Q_2 be the approximate solvers for each transport problem, that is $u^{n+1} = Q_1(\Delta t)u^n$ is an approximation of (3.37a), and $u^{n+1} = Q_2(\Delta t)u^n$ is an approximation of (3.37b). The splitting form is first order accurate in time if one evolves (3.37a) for a full time step, then evolves (3.37b) for a full time step, denoted as $u^{n+1} = Q_2(\Delta t) \cdot Q_1(\Delta t)u^n$. High order semi-discrete scheme for one step evolution from u^n to u^{n+1} is formulated in [20]. For example, the second order splitting, i.e., Strange splitting, is

$$u^{n+1} = Q_1(\frac{1}{2}\Delta t) \cdot Q_2(\Delta t) \cdot Q_1(\frac{1}{2}\Delta t)u^n. \quad (3.38)$$

The third order splitting is

$$u^{n+1} = Q_2(\Delta t) \cdot Q_1(-\frac{1}{24}\Delta t) \cdot Q_2(-\frac{2}{3}\Delta t) \cdot Q_1(\frac{3}{4}\Delta t) \cdot Q_2(\frac{2}{3}\Delta t) \cdot Q_1(\frac{7}{24}\Delta t)u^n. \quad (3.39)$$

And the forth order splitting is

$$\begin{aligned} u^{n+1} = & Q_2((\alpha + 1/2)\Delta t) \cdot Q_1((2\alpha + 1)\Delta t) \cdot Q_2(-\alpha\Delta t) \cdot Q_1(-(4\alpha + 1)\Delta t) \cdot \\ & Q_2(-\alpha\Delta t) \cdot Q_1((2\alpha + 1)\Delta t) \cdot Q_2((\alpha + 1/2)\Delta t)u^n, \end{aligned} \quad (3.40)$$

where $\alpha = (2^{1/3} + 2^{-1/3} - 1)/6$.

Note that the 1D1V VP system (1.1) can be split into two 1D advection equations:

$$f_t + vf(x) = 0, \quad (3.41a)$$

$$f_t + E(x, t)f_v = 0, \quad (3.41b)$$

known as spatial advection and velocity acceleration, respectively. Similarly, we can get the high order approximation by using (3.38), (3.39) or (3.40). Further, the electrostatic field $E(x, t)$ is obtained by solving Poisson's equation (1.1b) when it is needed.

4 Numerical Examples

In this section, we present the results of our numerical experiments for the schemes described in the previous sections. We will use the third and the fifth order WENO integrations for spatial discretization, denoted as WENO3 and WENO5. And the time discretization $RK(2, 3)$ and $RK(4, 4)$ are used for WENO3 and WENO5, respectively, to guarantee third order and fourth order accuracy in time. For 1D problems, the time steps are defined as

$$\Delta t = CFL \Delta x / |c|,$$

where $CFL = 1.5$ for $RK(2, 3)$ and $CFL = 2.9$ for $RK(4, 4)$. While for 2D problems, the time steps are chosen as

$$\Delta t = CFL / \max(|c_x|/\Delta x, |c_y|/\Delta y)$$

where $|c_x|$ and $|c_y|$ are the maximum wave propagation speeds in the x - and y -directions, and Δx and Δy are grid sizes in the x - and y -directions, respectively. For $RK(2, 3)$, we use the third order time splitting scheme (3.39) with $CFL = 1.5$. While, for $RK(4, 4)$, we use the forth order time splitting scheme (3.40) with $CFL = 1.6$.

To avoid the effect of rounding error, we change the condition (2.32) into

$$u_i^n - \frac{\Delta t}{\Delta x} (\tilde{f}_{i+1/2} - \tilde{f}_{i-1/2}) \geq 10^{-16}.$$

4.1 Basic 1-D and 2-D tests

Example 4.1 (one-dimensional linear advection equation)

$$u_t + u_x = 0, \quad x \in [-\pi, \pi] \tag{4.42}$$

with two different initial conditions:

- continuous initial condition $u(x, 0) = \cos(x)^4$,
- discontinuous initial condition: $u(x, 0) = \begin{cases} 1, & x \in [-\frac{1}{4}\pi, \frac{1}{4}\pi] \\ 0, & \text{otherwise.} \end{cases}$

The initial condition can be extended periodically on R with period 2π . The exact solution is $u(x, t) = u(x - t, 0)$. For the continuous problem, we test with a 2π -periodic boundary condition, a Dirichlet boundary condition $u(-\pi, t) = \cos(-\pi - t)^4$ and a Neumann boundary condition

$u_x(-\pi, t) = -4 \cos(-\pi - t)^3 \sin(-\pi - t)$. The L_1 and L_∞ errors and orders of accuracy are listed in Table 4.1, 4.2 and 4.3, and the PP-limiter is applied. We observe that WENO3 with $RK(2, 3)$ exhibits third order accuracy, and WENO5 with $RK(4, 4)$ exhibits the fourth order accuracy. The PP-limiter can keep the solutions non-negative without loss of accuracy. While for the discontinuous case (Figure 4.1), we consider the 2π -periodic boundary condition and the following Dirichlet boundary condition

$$u(-\pi, t) = \begin{cases} 1, & t \in [\frac{3}{4}\pi, \frac{5}{4}\pi], \\ 0, & t \in (0, \frac{3}{4}\pi) \cup (\frac{5}{4}\pi, 2\pi). \end{cases}$$

It is observed that the WENO methodology used effectively suppress the unphysical oscillations for both schemes.

Table 4.1: *Accuracy on the one-dimensional linear advection equation with continuous initial condition. $T = 2\pi$. Periodic boundary condition.*

		N_x	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	20	6.29E-01	—	2.36E-01	—	7.11E-03
		40	1.21E-01	2.38	4.29E-02	2.46	-7.30E-03
		80	1.32E-02	3.20	4.77E-03	3.17	-1.48E-03
		160	1.25E-03	3.39	5.21E-04	3.19	-2.32E-04
		320	1.17E-04	3.42	4.40E-05	3.57	-2.74E-05
		640	1.50E-05	2.96	4.83E-06	3.19	-2.89E-06
	With PP-limiters	20	6.28E-01	—	2.33E-01	—	1.35E-02
		40	1.22E-01	2.37	4.38E-02	2.41	1.11E-16
		80	1.29E-02	3.24	4.95E-03	3.15	1.11E-16
		160	1.19E-03	3.44	4.87E-04	3.35	1.11E-16
		320	1.17E-04	3.35	5.92E-05	3.04	1.11E-16
		640	1.50E-05	2.96	8.77E-06	2.76	1.11E-16
WENO5	Without PP-limiters	20	4.17E-01	—	1.57E-01	—	-4.28E-02
		40	3.63E-02	3.52	1.95E-02	3.01	-5.50E-03
		80	2.26E-03	4.01	1.32E-03	3.89	-4.79E-05
		160	1.24E-04	4.19	8.48E-05	3.96	-6.48E-06
		320	5.49E-06	4.50	2.05E-06	5.37	-3.83E-07
		640	3.26E-07	4.08	8.86E-08	4.53	-1.01E-09
	With PP-limiters	20	5.06E-01	—	2.00E-01	—	1.59E-02
		40	5.10E-02	3.31	2.93E-02	2.77	7.06E-04
		80	6.01E-03	3.08	4.86E-03	2.59	3.36E-04
		160	1.55E-04	5.28	1.28E-04	5.25	7.28E-06
		320	5.43E-06	4.84	1.75E-06	6.19	4.75E-08
		640	3.26E-07	4.06	8.86E-08	4.30	1.73E-10

Example 4.2 (Rigid body rotation)

$$u_t + yu_x - xu_y = 0, \quad x, y \in \Omega \quad (4.43)$$

with two different initial conditions:

Table 4.2: Accuracy on the one-dimensional linear advection equation with continuous initial condition. $T = 2\pi$. Dirichlet boundary condition.

		N_x	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	20	4.55E-01	–	3.11E-01	–	-4.42E-03
		40	9.09E-02	2.32	6.66E-02	2.22	-1.28E-03
		80	1.10E-02	3.05	1.15E-02	2.54	-3.71E-04
		160	1.08E-03	3.35	2.46E-03	2.22	-1.09E-04
		320	8.87E-05	3.60	1.95E-04	3.66	-1.82E-05
		640	9.63E-06	3.20	6.09E-06	5.00	-2.15E-06
	With PP-limiters	20	4.53E-01	–	3.11E-01	–	1.11E-16
		40	9.07E-02	2.32	6.66E-02	2.22	5.55E-17
		80	1.10E-02	3.04	1.15E-02	2.54	5.55E-17
		160	1.06E-03	3.38	2.46E-03	2.22	5.55E-17
		320	8.87E-05	3.58	1.95E-04	3.66	1.11E-16
		640	9.63E-06	3.20	6.61E-06	4.88	5.55E-17
WENO5	Without PP-limiters	20	3.59E-01	–	2.14E-01	–	1.18E-02
		40	4.37E-02	3.04	3.24E-02	2.73	9.44E-04
		80	2.65E-03	4.05	5.79E-03	2.48	1.98E-05
		160	1.08E-04	4.61	1.69E-04	5.10	-1.52E-06
		320	2.91E-06	5.22	1.70E-06	6.64	-1.99E-07
		640	1.69E-07	4.10	8.22E-08	4.37	6.37E-09
	With PP-limiters	20	3.58E-01	–	2.15E-01	–	1.18E-02
		40	4.37E-02	3.04	3.24E-02	2.73	1.29E-03
		80	2.83E-03	3.95	5.79E-03	2.48	3.53E-05
		160	1.18E-04	4.57	1.75E-04	5.05	6.26E-06
		320	2.89E-06	5.36	1.67E-06	6.71	1.11E-16
		640	1.69E-07	4.10	8.22E-08	4.35	6.37E-09

- continuous initial condition, with $\Omega = [-\frac{1}{2}\pi, \frac{1}{2}\pi]^2$

$$u(x, y, 0) = 0.5B(\sqrt{x^2 + 8y^2}) + 0.5B(\sqrt{8x^2 + y^2}),$$

$$\text{where } B(r) = \begin{cases} \cos(r)^6, & \text{if } r \leq \frac{1}{2}\pi, \\ 0, & \text{otherwise.} \end{cases}$$

- dicontinuous initial condition, with $\Omega = [-1, 1]^2$

$$u(x, y, 0) = \begin{cases} 1, & (x, y) \in [-0.75, 0.75] \times [-0.25, 0.25] \cup [-0.25, 0.25] \times [-0.75, 0.75], \\ 0, & \text{otherwise.} \end{cases}$$

We simulate the problems with the zero Dirichlet boundary conditions for both problems. In Table 4.4, we summarize the convergence study and minimum solution values at the final time $T = 2\pi$. It is observed that our schemes can achieve the designed order of accuracy and meanwhile the PP-limiter is able to keep positivity without destroying the accuracy of the original scheme. Furthermore, similar to 1D case, WENO methodology removes unphysical oscillations as expected (Figure 4.2).

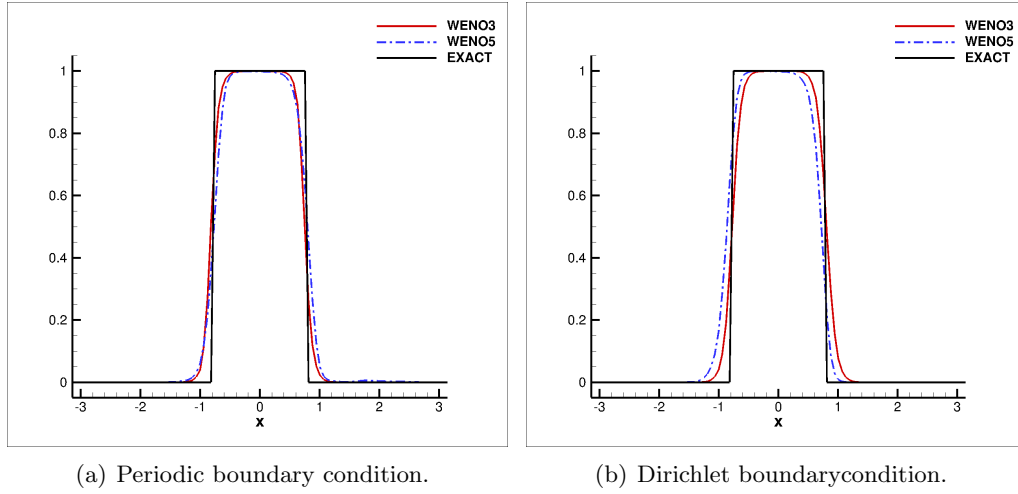


Figure 4.1: *The one-dimensional linear advection equation with discontinuous initial condition. $T = 2\pi$. 100 grid points.*

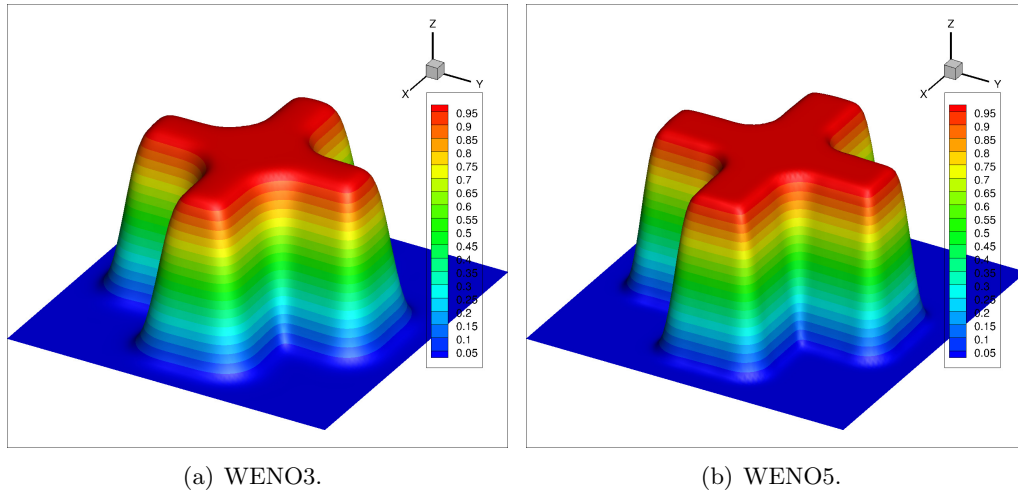


Figure 4.2: *The rigid body rotation with discontinuous initial condition. $T = 2\pi$. 100×100 grid points.*

Table 4.3: *Accuracy on the one-dimensional linear advection equation with continuous initial condition. $T = 2\pi$. Neumann boundary condition.*

		N_x	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	20	4.52E-01	–	3.09E-01	–	-7.98E-03
		40	9.14E-02	2.31	6.65E-02	2.22	-1.22E-03
		80	1.11E-02	3.05	1.15E-02	2.54	-4.02E-04
		160	1.09E-03	3.34	2.46E-03	2.22	-1.10E-04
		320	9.09E-05	3.58	1.95E-04	3.66	-1.83E-05
		640	9.93E-06	3.20	6.09E-06	5.00	-2.16E-06
		20	4.50E-01	–	3.09E-01	–	1.11E-16
	With PP-limiters	40	9.12E-02	2.30	6.64E-02	2.22	5.55E-17
		80	1.12E-02	3.04	1.15E-02	2.54	5.55E-17
		160	1.07E-03	3.37	2.46E-03	2.22	5.55E-17
		320	9.10E-05	3.56	1.95E-04	3.66	1.11E-16
		640	9.93E-06	3.20	6.65E-06	4.87	5.55E-17
WENO5	Without PP-limiters	20	3.92E-01	–	2.22E-01	–	-2.93E-02
		40	4.75E-02	3.05	3.18E-02	2.80	-1.87E-03
		80	5.02E-03	3.24	5.74E-03	2.47	-4.70E-05
		160	3.13E-04	4.01	6.20E-04	3.21	9.64E-06
		320	4.11E-06	6.25	1.91E-06	8.34	-4.50E-07
		640	2.51E-07	4.03	1.20E-07	4.00	-1.00E-08
		20	3.95E-01	–	2.45E-01	–	4.85E-03
	With PP-limiters	40	5.74E-02	2.78	4.93E-02	2.31	0.00E+00
		80	6.32E-03	3.18	6.80E-03	2.86	1.56E-04
		160	2.35E-04	4.75	4.00E-04	4.09	1.28E-05
		320	3.93E-06	5.90	1.91E-06	7.71	1.11E-16
		640	2.29E-07	4.10	1.20E-07	4.00	0.00E+00

4.2 The VP systems

In this subsection, we consider the Vlasov-Poisson equation (1.1) with the following four initial conditions:

- Strong Landau damping:

$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha \cos(kx)) \exp(-\frac{v^2}{2}), x \in [0, L], v \in [-V_c, V_c]$$

where $\alpha = 0.5$, $k = 0.5$, $L = 4\pi$ and $V_c = 2\pi$.

- Two-stream instability I:

$$f(x, v, 0) = \frac{2}{7\sqrt{2\pi}}(1 + 5v^2)(1 + \alpha((\cos(2kx) + \cos(3kx))/1.2 + \cos(kx))) \exp(-\frac{v^2}{2}),$$

$$x \in [0, L], v \in [-V_c, V_c]$$

where $\alpha = 0.01$, $k = 0.5$, $L = 4\pi$ and $V_c = 2\pi$.

Table 4.4: Accuracy on the rigid body rotation with continuous initial condition. $T = 2\pi$.

		$N_x \times N_y$	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	20×20	1.84E-01	–	4.25E-01	–	-2.22E-05
		40×40	9.84E-02	0.90	1.72E-01	1.31	-1.41E-04
		80×80	2.66E-02	1.89	3.35E-02	2.36	-2.46E-04
		160×160	3.41E-03	2.96	4.40E-03	2.93	-2.84E-04
		320×320	3.96E-04	3.10	5.20E-04	3.08	-7.25E-05
	With PP-limiters	20×20	1.84E-01	–	4.25E-01	–	0.00E+00
		40×40	9.84E-02	0.90	1.72E-01	1.31	0.00E+00
		80×80	2.65E-02	1.89	3.35E-02	2.36	0.00E+00
		160×160	3.32E-03	3.00	4.40E-03	2.93	0.00E+00
		320×320	3.92E-04	3.09	5.20E-04	3.08	0.00E+00
WENO5	Without PP-limiters	20×20	1.43E-01	–	3.32E-01	–	-9.51E-07
		40×40	6.08E-02	1.24	7.43E-02	2.16	-1.67E-05
		80×80	4.98E-03	3.61	7.49E-03	3.31	-2.50E-05
		160×160	1.34E-04	5.22	2.75E-04	4.77	-4.76E-06
		320×320	3.73E-06	5.16	7.56E-06	5.18	-1.03E-07
	With PP-limiters	20×20	1.43E-01	–	3.32E-01	–	0.00E+00
		40×40	6.09E-02	1.24	7.43E-02	2.16	0.00E+00
		80×80	4.98E-03	3.61	7.49E-03	3.31	0.00E+00
		160×160	1.34E-04	5.22	2.75E-04	4.77	0.00E+00
		320×320	3.72E-06	5.17	7.56E-06	5.18	0.00E+00

- Two-stream instability II:

$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha \cos(kx))v^2 \exp(-\frac{v^2}{2}), \quad x \in [0, L], \quad v \in [-V_c, V_c]$$

where $\alpha = 0.05$, $k = 0.5$, $L = 4\pi$ and $V_c = 2\pi$.

- Bump-on-tail instability:

$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha \cos(kx))(0.9 \exp(-0.5v^2) + 0.2 \exp(-4(v - 4.5)^2))$$

$$x \in [-L, L], \quad v \in [-V_c, V_c]$$

where $\alpha = 0.04$, $k = 0.3$, $L = \frac{10}{3}\pi$ and $V_c = 10$.

In our numerical tests, periodic boundary conditions are imposed in both directions, and a fast Fourier transform (FFT) is used to solve the 1-D Poisson equation. The VP system is time-reversible, meaning that if the solution f is first evolved to time $T/2$, at when we reverse the solution in v -direction, i.e., $\tilde{f}(x, v, \frac{1}{2}T) = f(x, -v, \frac{1}{2}T)$, and continue to evolve the reserved solution \tilde{f} to time T , then we can recover the initial condition with reverse velocity field, i.e., $\tilde{f}(x, v, T) = f(x, -v, 0)$. Below, we make use of such a property of the VP system to perform the accuracy test. In Table 4.5-4.8, we list the errors of $\tilde{f}(x, -v, T)$ and associated orders of accuracy at time $T = 10$, without

or with the PP-limiter. As expected, all schemes achieve the designed order accuracy and the limiter does not adversely affect the accuracy.

Below, we briefly recall some classical preservation results in the VP system. We hope that our numerical solutions can preserve these classical conserved quantities as much as possible.

$$\begin{aligned}
Mass &= \int_v \int_x f(x, v, t) dx dv, \\
\|f\|_{L_p} &= (\int_v \int_x |f(x, v, t)|^p dx dv)^{1/p}, \quad 1 \leq p < \infty, \\
Energy &= \frac{1}{2} (\int_v \int_x f(x, v, t) v^2 dx dv + \int_x E^2(x, t) dx), \\
Momentum &= \int_v \int_x f(x, v, t) v dx dv,
\end{aligned}$$

For all problems, the WENO5 scheme performs better than the WENO3 scheme, showed in Figure 4.3-4.11. It is observed that both schemes preserve L_1 norm (Figure 4.13) up to the machine error, since the scheme are both conservative and positivity-preserving if the PP-limiter is applied. While the WENO5 scheme does a better job in preserving the energy, L_2 norm and momentum than the WENO3 scheme.

We test Two-stream instability I with second order central difference scheme in time, which is non-conditionally stable and conditionally SSP. CFL is taken as 10 such that the scheme is stable but not SSP. In space, we use WENO5. In Figure 4.7, we can observe that the scheme simulates the structures well. We remark that the SSP property is not always desired for the VP simulations, especially when the solution is smooth.

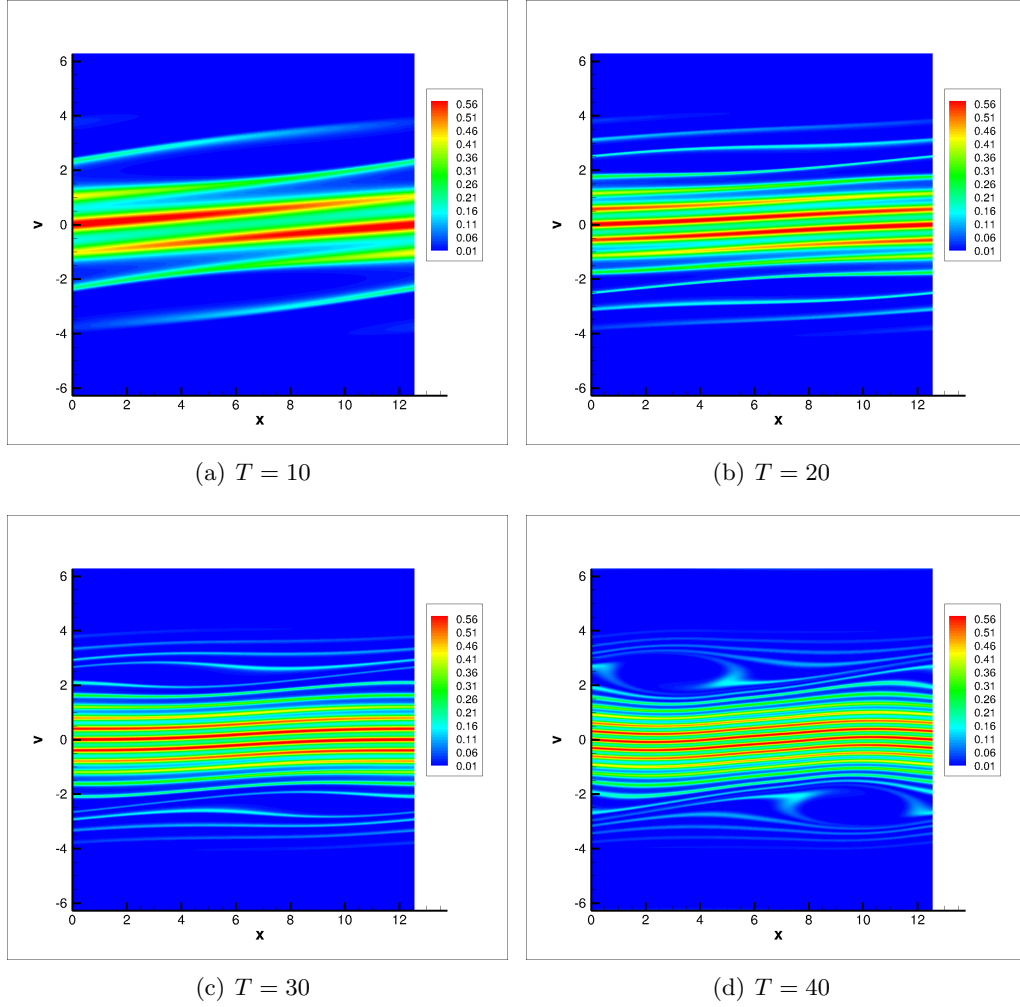


Figure 4.3: *Strong Landau Damping. WENO3, with 256×1024 grid points.*

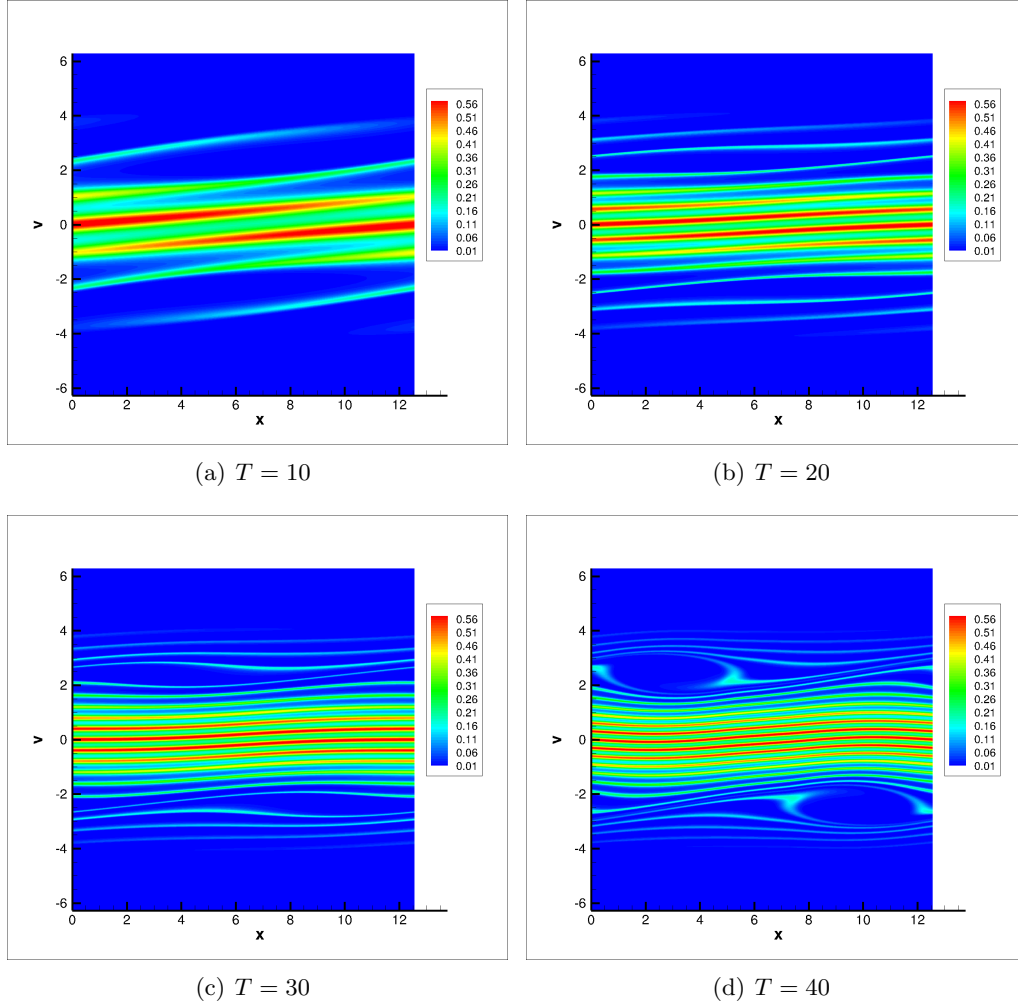


Figure 4.4: *Strong Landau Damping. WENO5, with 256×1024 grid points.*

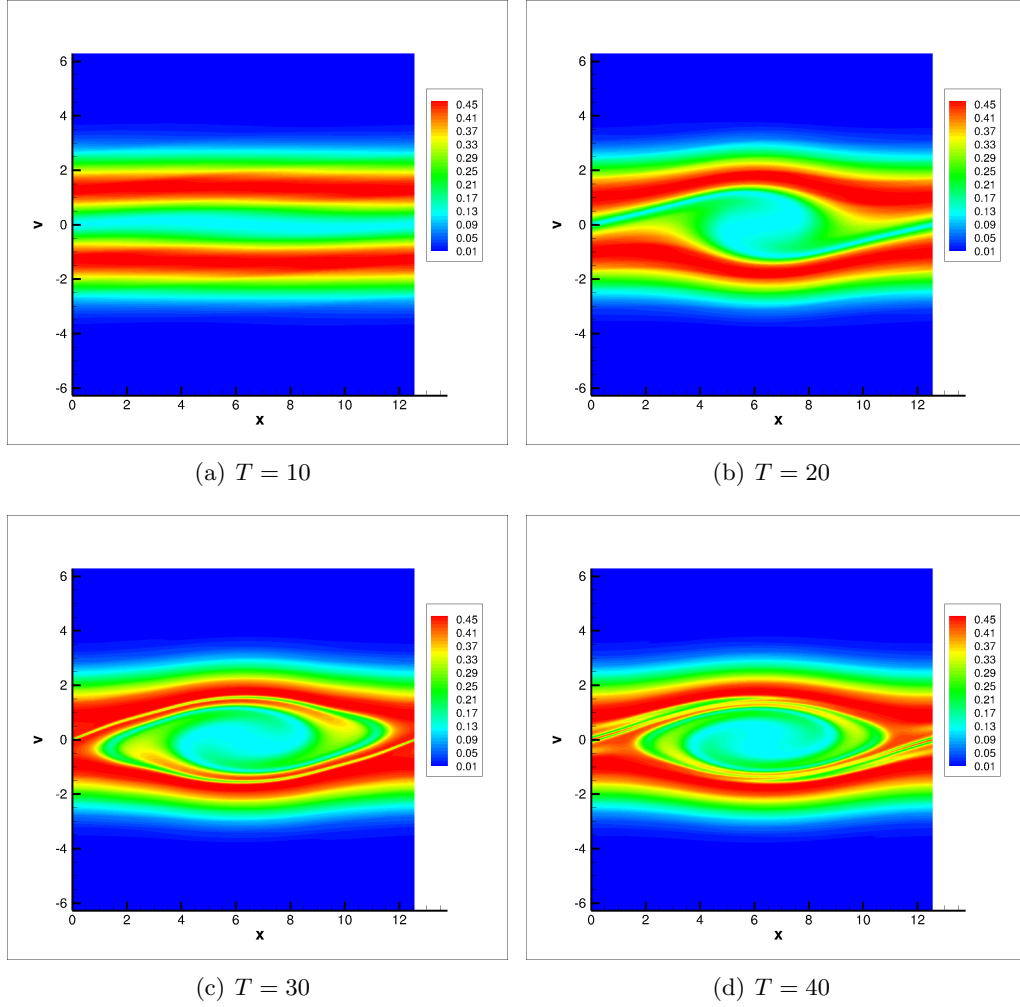


Figure 4.5: *Two-stream instability I. WENO3, with 256×1024 grid points.*

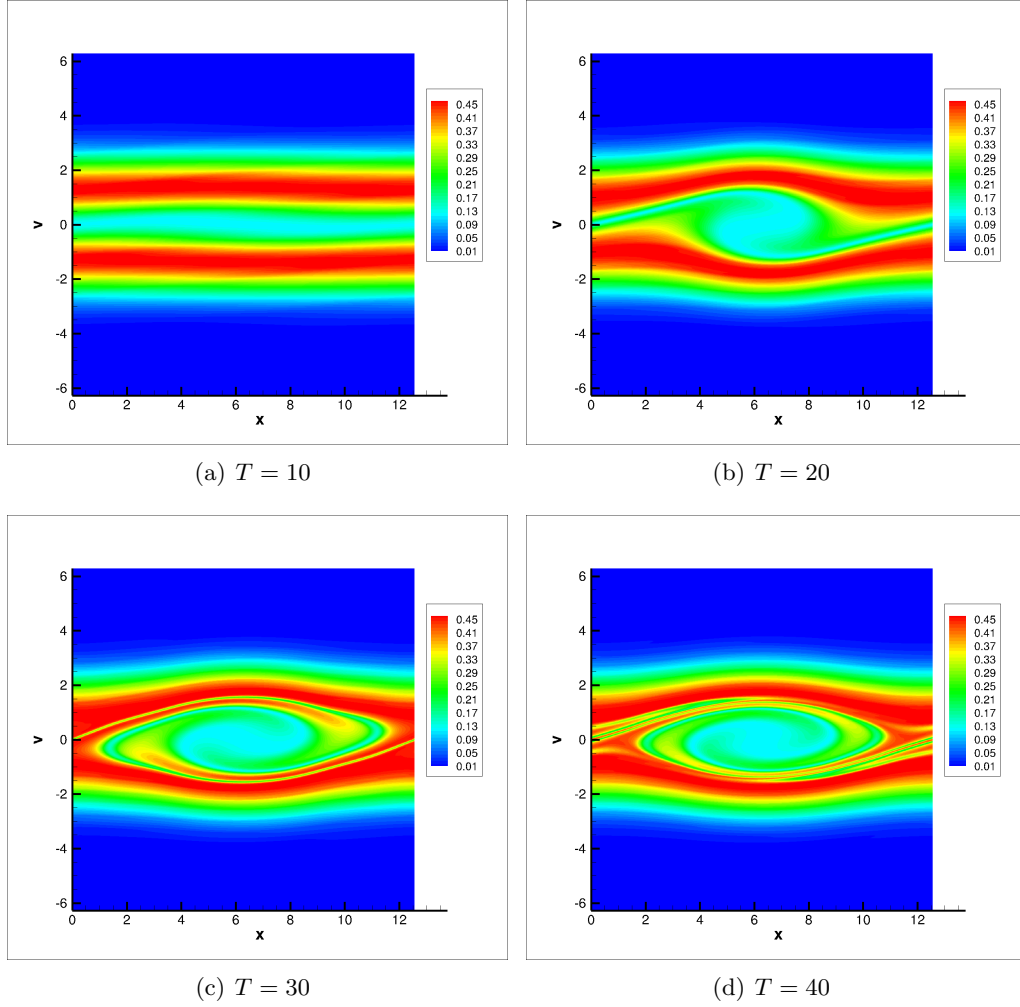


Figure 4.6: *Two-stream instability I. WENO5, with 256×1024 grid points.*

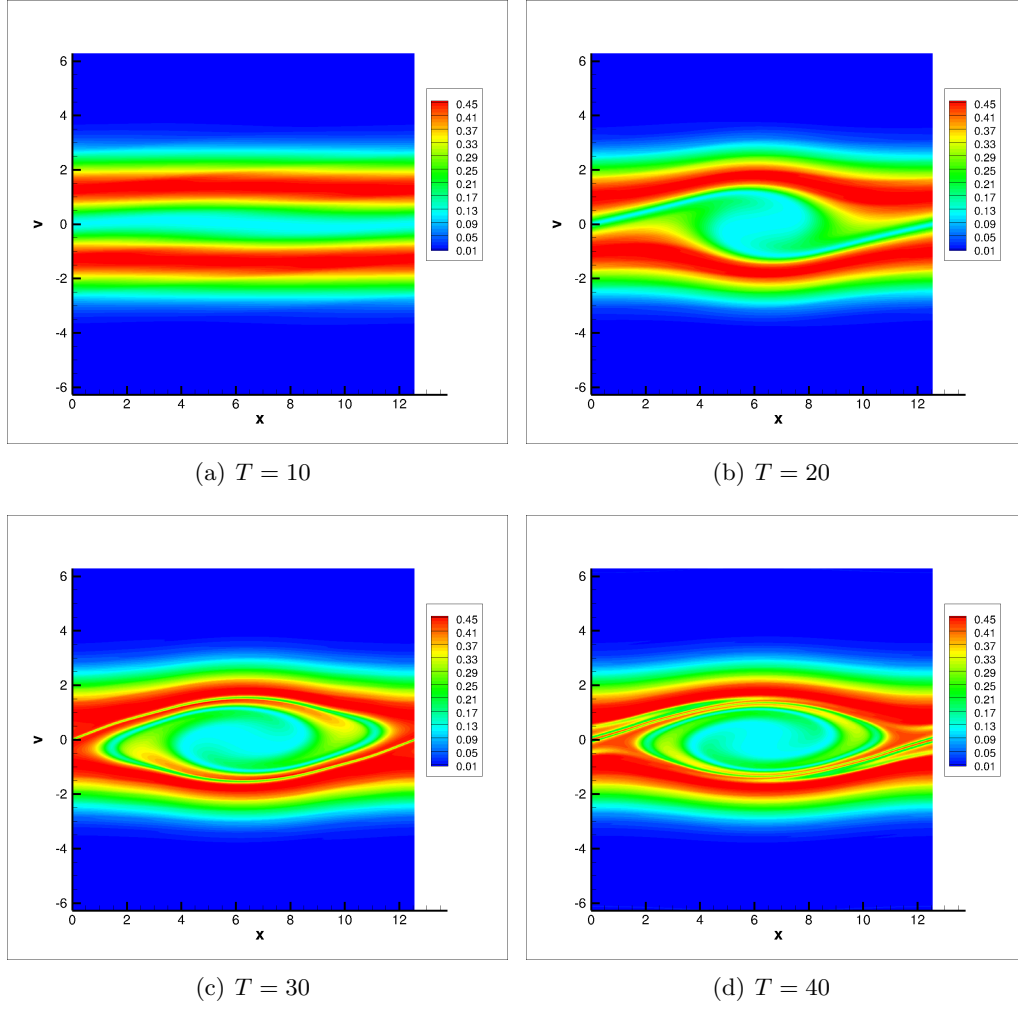


Figure 4.7: *Two-stream instability I. Second order centel difference scheme with WENO5. 256×1024 grid points.*

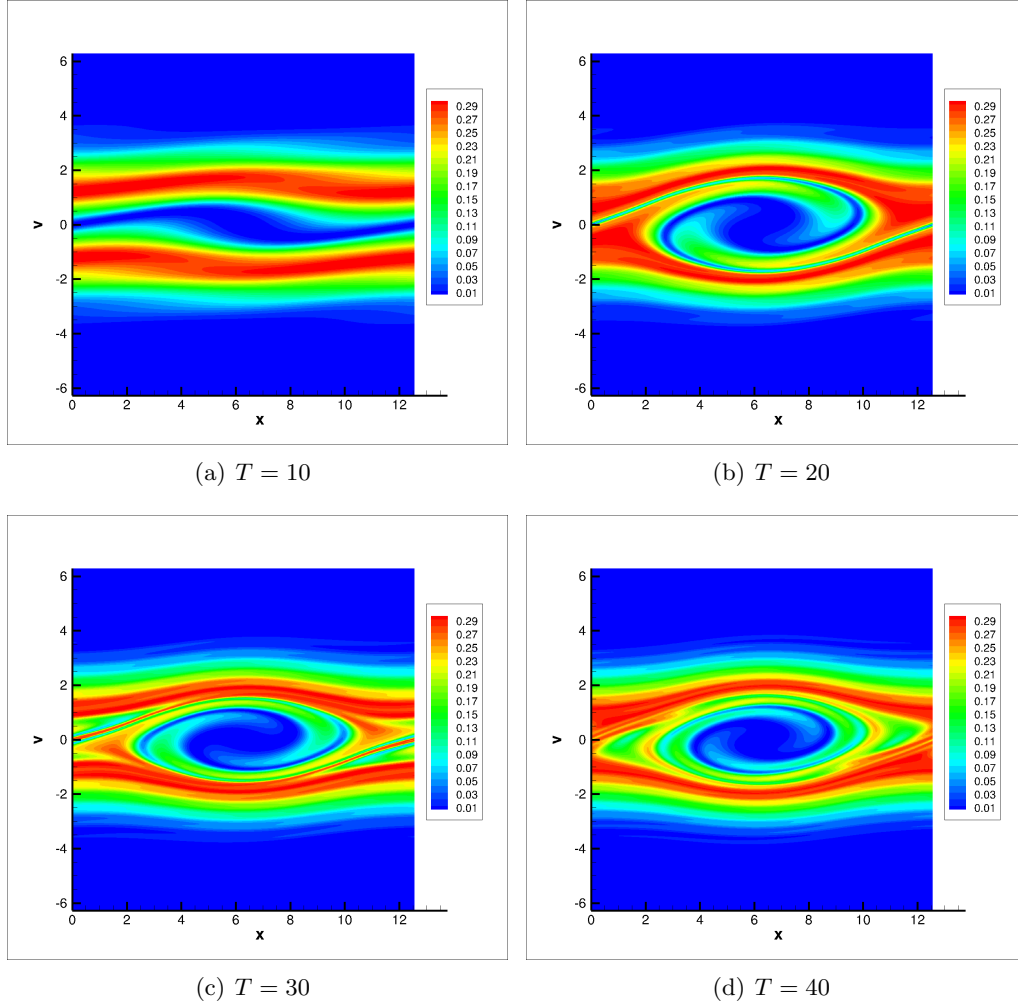


Figure 4.8: *Two-stream instability II. WENO3, with 256×1024 grid points.*

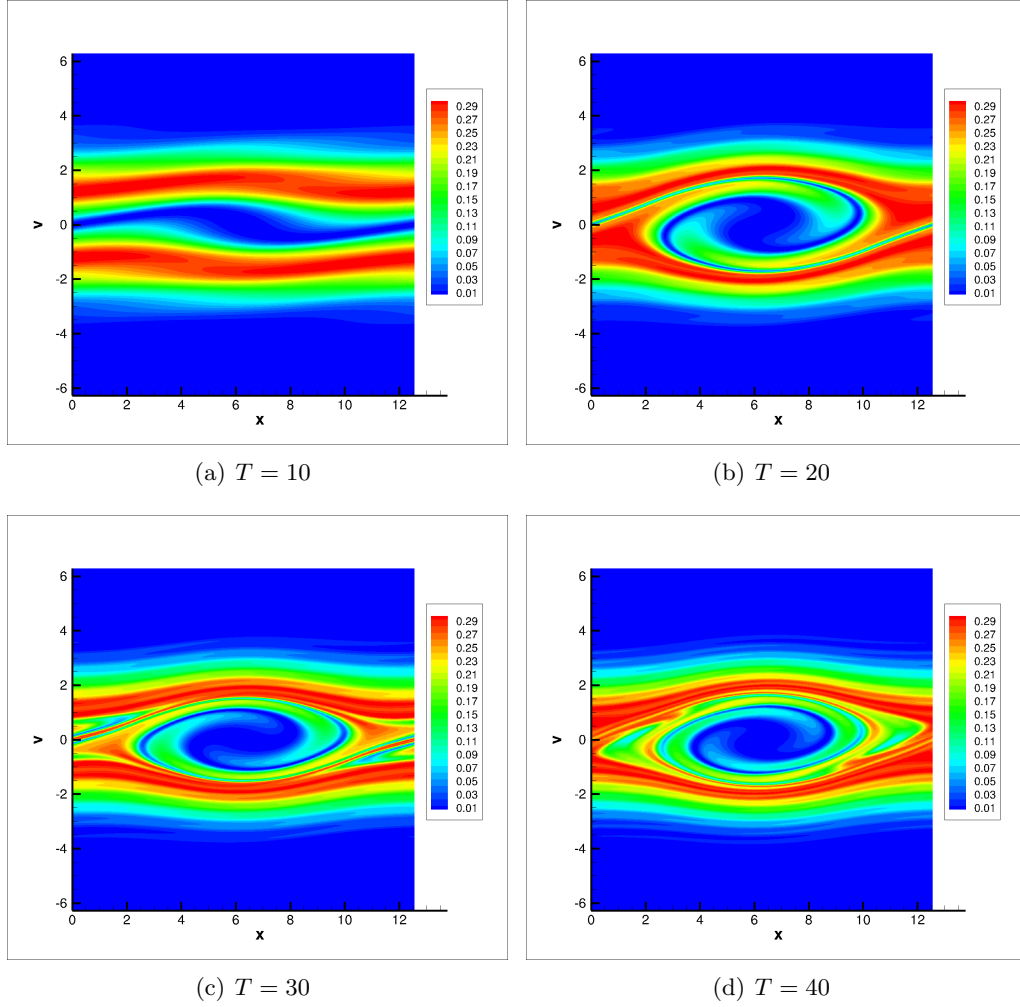


Figure 4.9: *Two-stream instability II. WENO5, with 256×1024 grid points.*

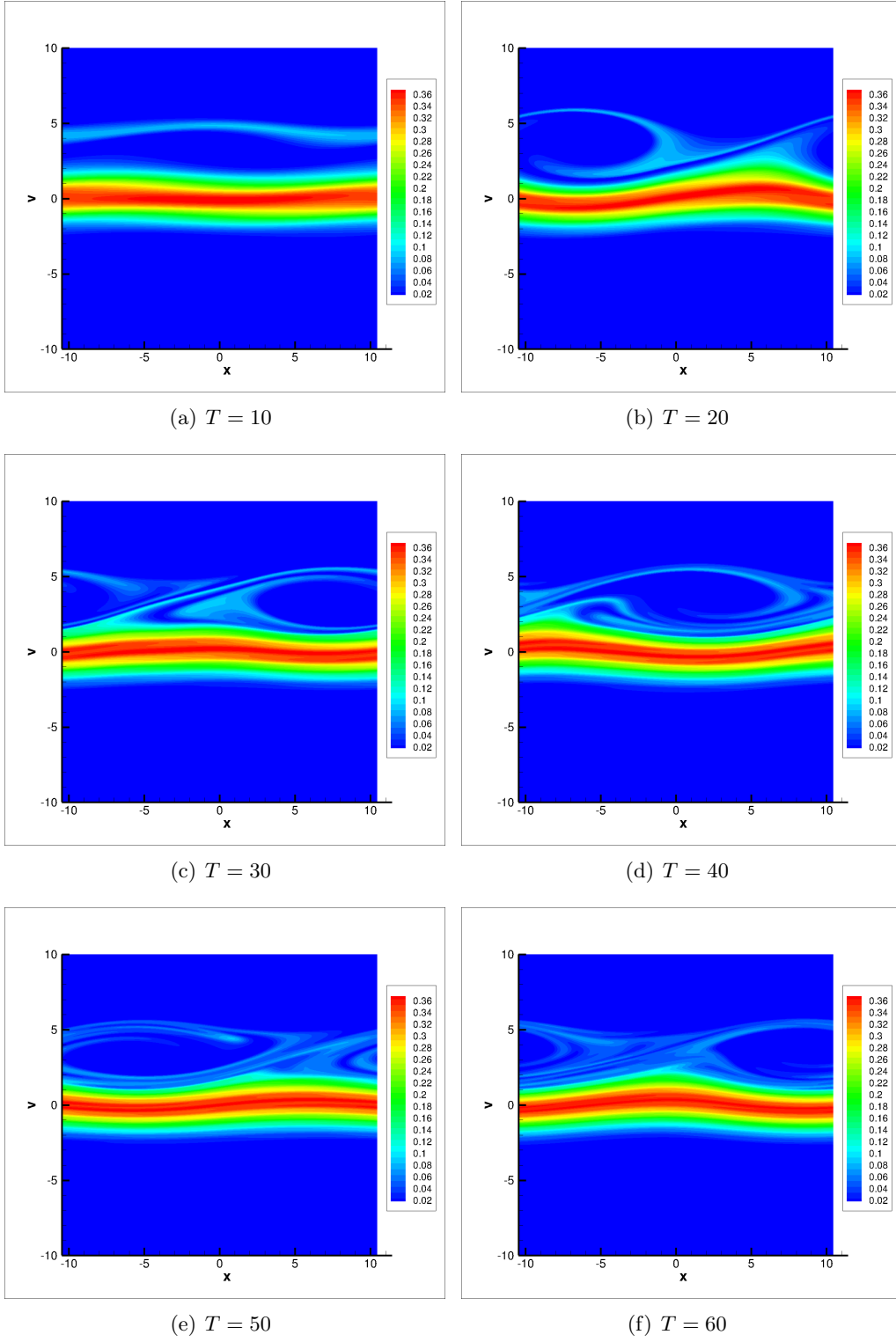


Figure 4.10: *Bump-on-tail instability. WENO3, with 256×1024 grid points.*

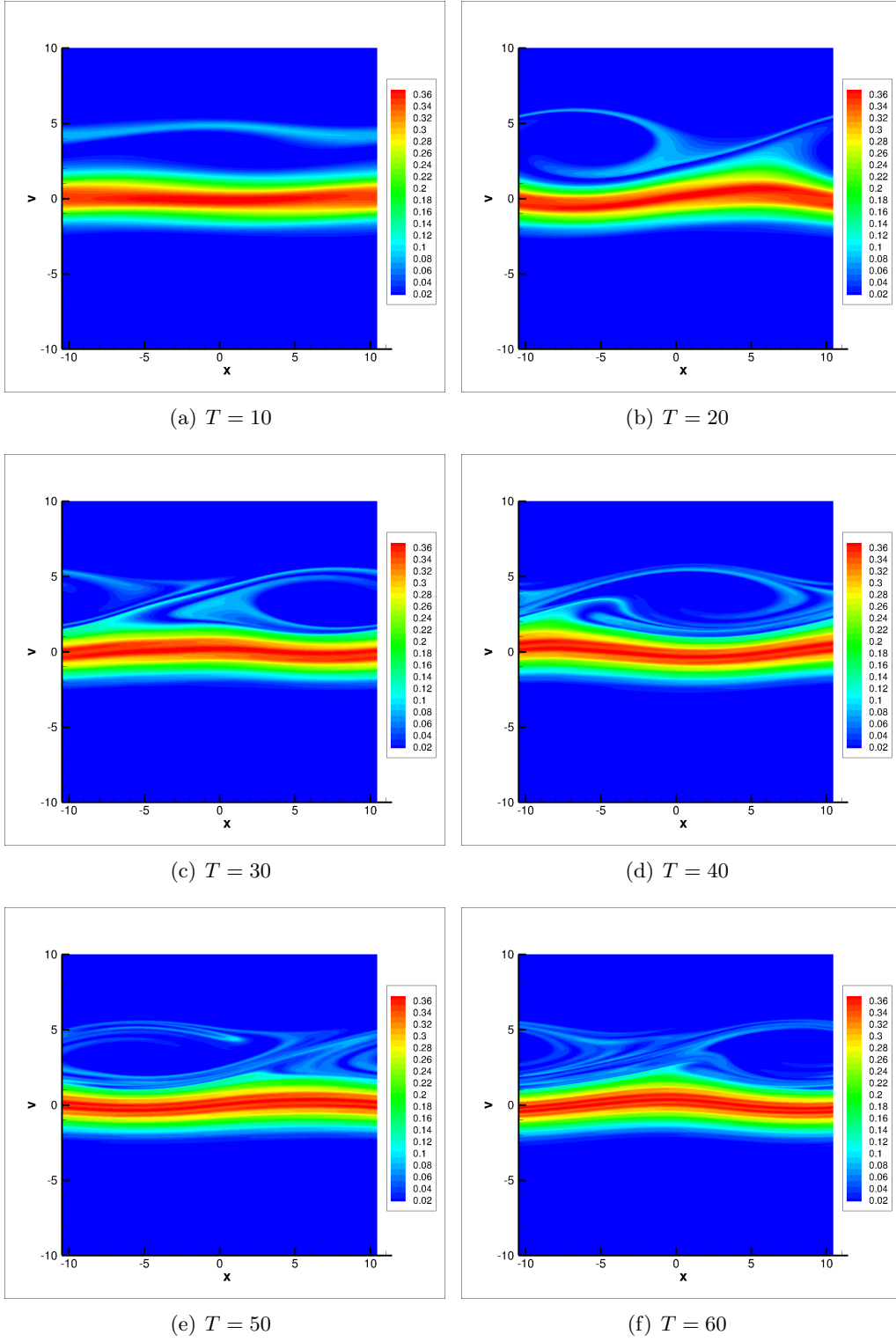


Figure 4.11: *Bump-on-tail instability. WENO5, with 256×1024 grid points.*

Table 4.5: *Accuracy on strong Landau damping. $T = 10$.*

		$N_x \times N_v$	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	32×64	1.49E+00	–	8.98E-02	–	-1.00E-04
		64×128	3.04E-01	2.30	2.37E-02	1.92	-8.67E-05
		128×256	3.78E-02	3.01	3.56E-03	2.73	-3.48E-05
		256×512	4.43E-03	3.09	4.05E-04	3.14	-5.84E-06
	With PP-limiters	32×64	1.49E+00	–	8.98E-02	–	9.99E-17
		64×128	3.03E-01	2.30	2.37E-02	1.92	1.00E-16
		128×256	3.78E-02	3.01	3.56E-03	2.73	1.00E-16
		256×512	4.44E-03	3.09	4.05E-04	3.14	1.00E-16
WENO5	Without PP-limiters	32×64	5.70E-01	–	4.28E-02	–	-1.79E-04
		64×128	4.32E-02	3.72	5.02E-03	3.09	-1.47E-05
		128×256	1.46E-03	4.89	1.87E-04	4.75	-2.17E-06
		256×512	3.98E-05	5.19	4.82E-06	5.28	-1.38E-07
	With PP-limiters	32×64	5.70E-01	–	4.28E-02	–	1.00E-16
		64×128	4.31E-02	3.72	5.02E-03	3.09	1.00E-16
		128×256	1.46E-03	4.88	1.87E-04	4.74	1.00E-16
		256×512	4.01E-05	5.19	4.83E-06	5.28	1.00E-16

Table 4.6: *Accuracy on two-stream instability I. $T = 10$.*

		$N_x \times N_v$	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	32×64	7.17E-02	–	3.56E-03	–	-3.47E-06
		64×128	9.39E-03	2.93	4.03E-04	3.14	-3.87E-06
		128×256	1.05E-03	3.17	4.51E-05	3.16	-4.88E-07
		256×512	1.27E-04	3.05	5.44E-06	3.05	1.45E-08
	With PP-limiters	32×64	7.18E-02	–	3.56E-03	–	1.00E-16
		64×128	9.38E-03	2.94	4.03E-04	3.14	1.00E-16
		128×256	1.05E-03	3.16	4.51E-05	3.16	1.00E-16
		256×512	1.27E-04	3.05	5.44E-06	3.05	1.45E-08
WENO5	Without PP-limiters	32×64	3.27E-02	–	1.44E-03	–	2.58E-07
		64×128	1.13E-03	4.86	5.75E-05	4.65	-1.21E-07
		128×256	2.51E-05	5.49	1.84E-06	4.97	8.42E-08
		256×512	5.61E-07	5.49	4.49E-08	5.35	6.75E-08
	With PP-limiters	32×64	3.27E-02	–	1.44E-03	–	2.58E-07
		64×128	1.13E-03	4.86	5.75E-05	4.65	1.00E-16
		128×256	2.51E-05	5.49	1.84E-06	4.97	8.42E-08
		256×512	5.61E-07	5.49	4.49E-08	5.35	6.75E-08

Table 4.7: *Accuracy on two-stream instability II. $T = 10$.*

		$N_x \times N_v$	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	32×64	8.90E-02	–	4.56E-03	–	-9.96E-06
		64×128	9.40E-03	3.24	5.06E-04	3.17	-6.65E-06
		128×256	1.05E-03	3.16	5.94E-05	3.09	-8.71E-07
		256×512	1.22E-04	3.10	7.06E-06	3.07	-2.15E-08
	With PP-limiters	32×64	8.92E-02	–	4.56E-03	–	1.00E-16
		64×128	9.38E-03	3.25	5.06E-04	3.17	1.00E-16
		128×256	1.05E-03	3.16	5.94E-05	3.09	1.00E-16
		256×512	1.22E-04	3.10	7.06E-06	3.07	1.00E-16
WENO5	Without PP-limiters	32×64	7.58E-03	–	6.37E-04	–	-3.94E-07
		64×128	2.11E-04	5.17	2.36E-05	4.76	-9.53E-07
		128×256	5.29E-06	5.32	7.44E-07	4.99	1.70E-14
		256×512	1.43E-07	5.21	2.27E-08	5.04	2.43E-17
	With PP-limiters	32×64	7.66E-03	–	6.37E-04	–	1.00E-16
		64×128	2.17E-04	5.14	2.36E-05	4.76	1.00E-16
		128×256	5.31E-06	5.36	7.44E-07	4.99	5.24E-15
		256×512	1.43E-07	5.21	2.27E-08	5.04	1.24E-16

Table 4.8: *Accuracy on bump-on-tail instability. $T = 10$.*

		$N_x \times N_v$	L_1 errors	order	L_∞ error	order	min value
WENO3	Without PP-limiters	32×64	5.78E-01	–	1.97E-02	–	-1.28E-05
		64×128	1.10E-01	2.39	6.36E-03	1.63	-1.68E-05
		128×256	1.26E-02	3.13	8.16E-04	2.96	-5.76E-07
		256×512	1.44E-03	3.13	9.02E-05	3.18	-5.35E-07
	With PP-limiters	32×64	5.78E-01	–	1.97E-01	–	1.00E-16
		64×128	1.10E-01	2.39	6.36E-03	1.63	1.00E-16
		128×256	1.26E-02	3.13	8.16E-04	2.96	1.00E-16
		256×512	1.44E-03	3.13	9.02E-05	3.18	1.00E-16
WENO5	Without PP-limiters	32×64	2.25E-01	–	1.02E-02	–	-1.23E-06
		64×128	1.66E-02	3.76	1.44E-03	2.83	-1.19E-06
		128×256	7.49E-04	4.47	7.75E-05	4.21	-1.31E-06
		256×512	2.08E-05	5.17	2.79E-06	4.80	-1.48E-07
	With PP-limiters	32×64	2.25E-01	–	1.02E-02	–	1.00E-16
		64×128	1.66E-02	3.76	1.44E-03	2.83	1.00E-16
		128×256	7.46E-04	4.47	7.75E-05	4.21	1.00E-16
		256×512	2.09E-05	5.16	2.79E-06	4.80	1.00E-16

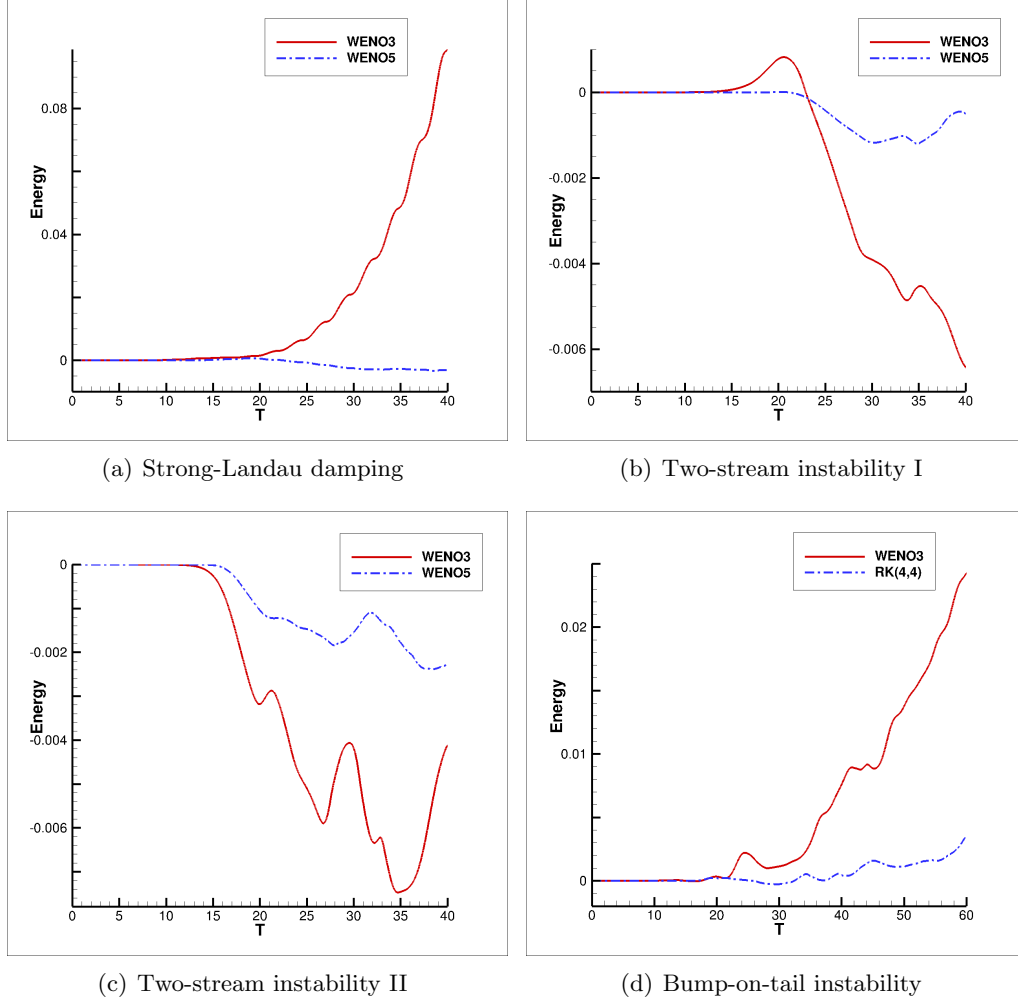


Figure 4.12: *The time evolution of the relative deviation in total energy. 256×1024 grid points.*

4.3 Plasma sheath

Next, we consider the so-called plasma sheath problem, which describes VP system with non-periodic boundary condition in x -direction. The initial condition is

$$f(x, v, 0) = \frac{1}{\sqrt{\alpha\pi}} \exp\left(-\frac{1}{\alpha}v^2\right), \quad x \in [0, L], \quad v \in [-V_c, V_c] \quad (4.44)$$

where $\alpha = 0.0005526350206$, $L = 1$ and $V_c = 0.2$. We assume that no electrons can enter the domain, meaning that we take zero-inflow boundary condition:

$$f(x = 0, t, v \geq 0) = f(x = L, t, v \leq 0) = 0. \quad (4.45)$$

The electric potential is fixed to a constant value at $x = 0$ and $x = L$. Without loss of generality, we take $\phi(x = 0, t) = \phi(x = L, t) = 0$. The solutions at the final time $T = 140$ are shown in Figure

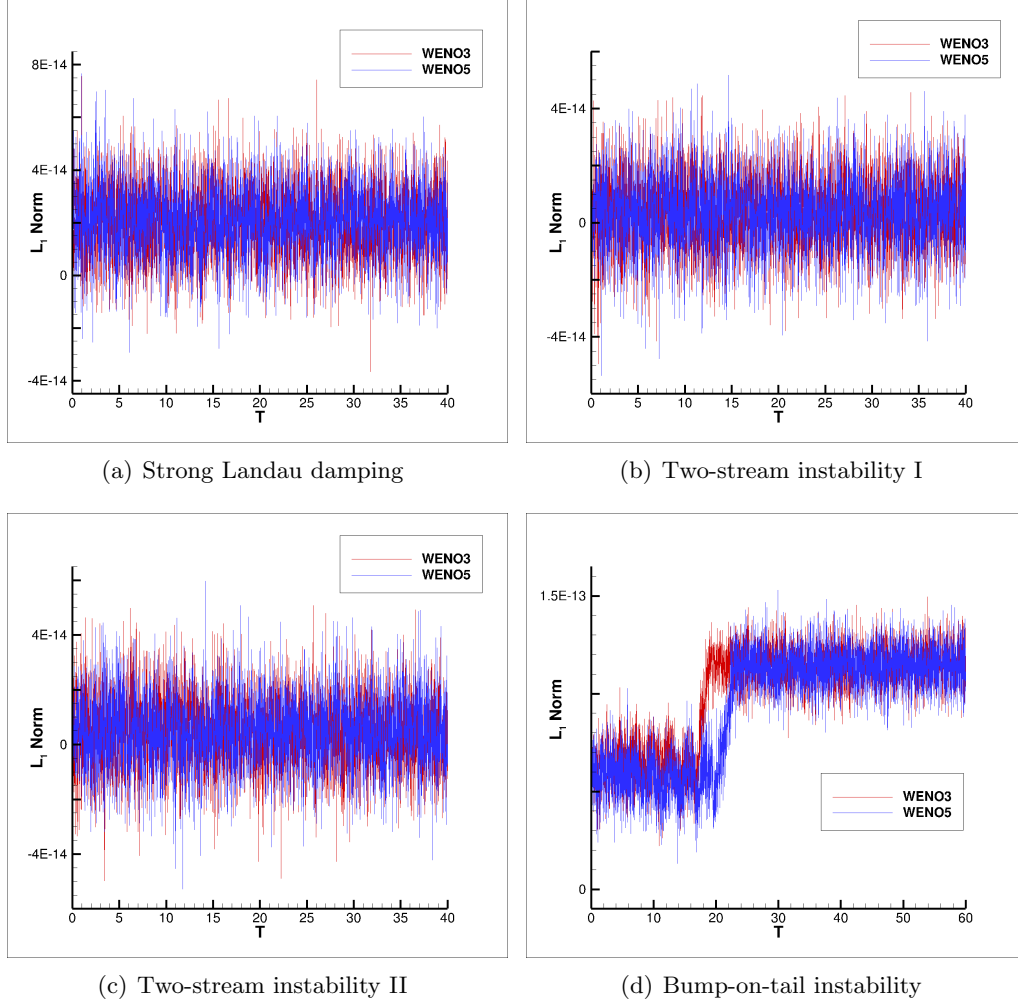


Figure 4.13: *The time evolution of the relative deviation in L_1 norm. 256×1024 grid points.*

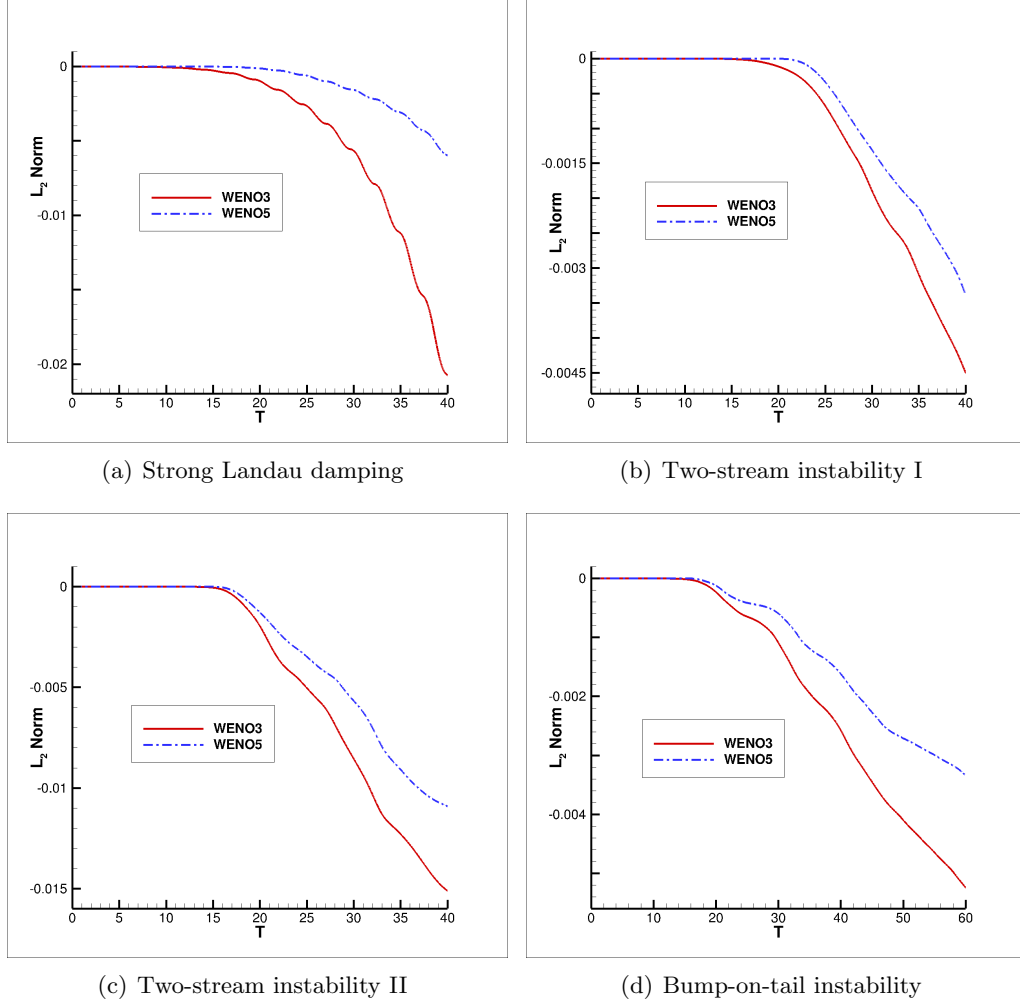


Figure 4.14: *The time evolution of the relative deviation in L_2 norm. 256×1024 grid points.*

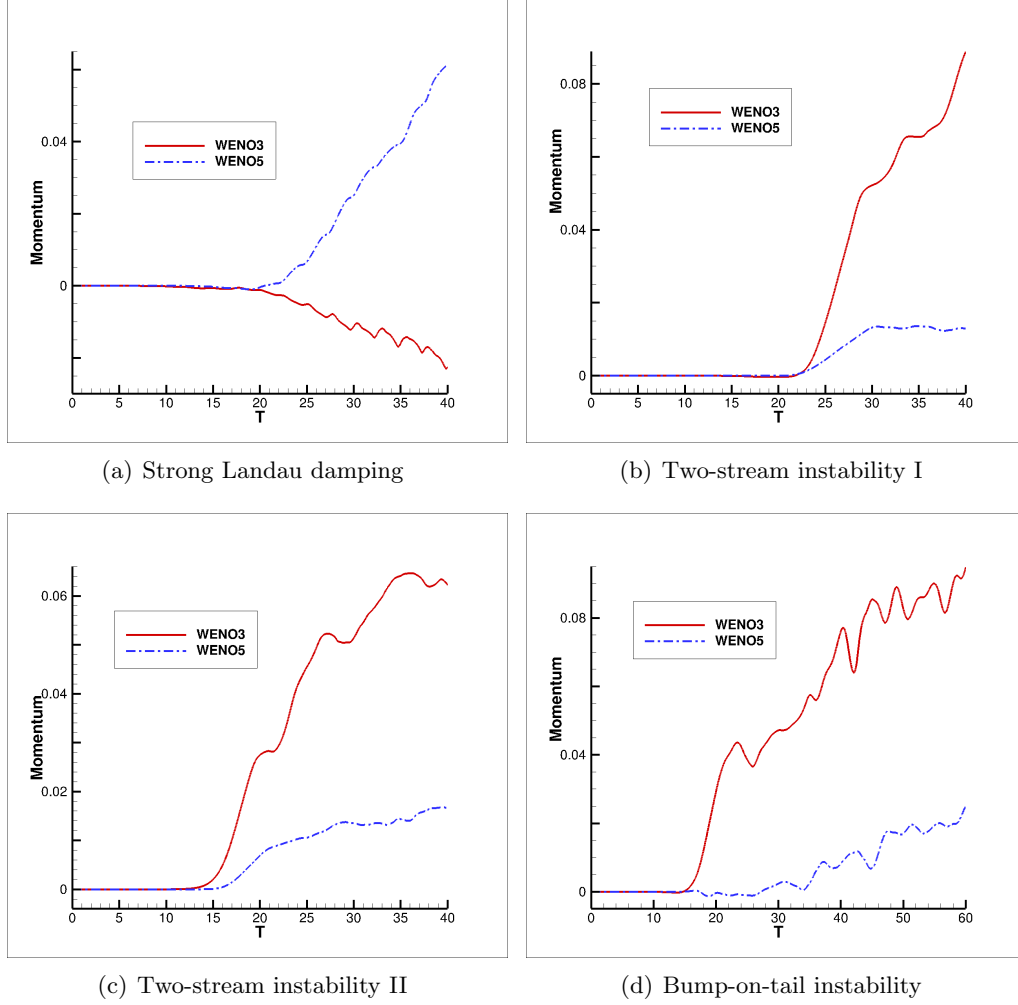
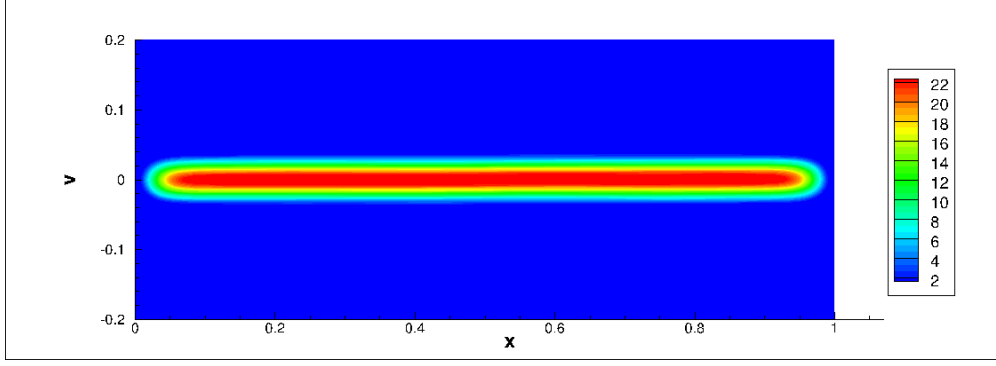
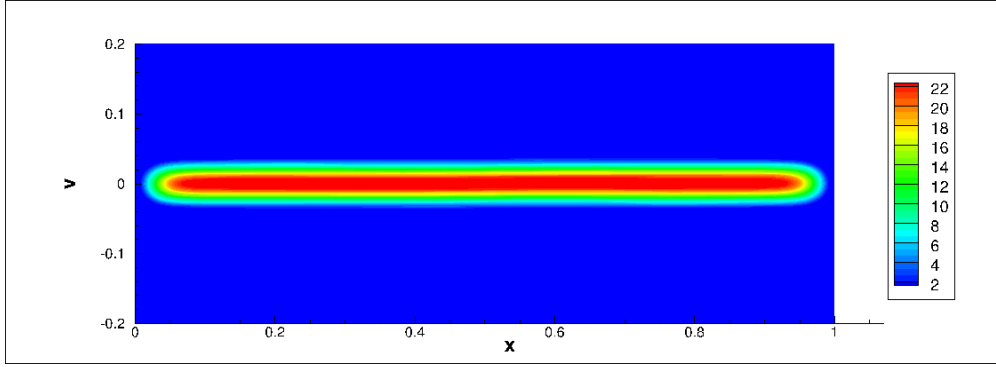


Figure 4.15: *The time evolution of momentum. 256×1024 grid points.*



(a) WENO3



(b) WENO5

Figure 4.16: *Plasma sheath*. 256×1024 grid points. $T = 140$

4.16, with 256×1024 grid points. Our results are consistent with that in [37] .

5 Conclusion

In this paper, we proposed a high order implicit Method of Lines Transpose (MOL^T) method for linear advection equations, which was further applied to the Vlasov-Poisson (VP) system for plasma simulations. The scheme coupled a WENO methodology and the diagonally implicit strong-stability-preserving Runge-Kutta method in the MOL^T framework, in order to attain high order accuracy in smooth region and avoid oscillation near the discontinuities with large time step evolution. Further, the scheme is able to efficiently deal with general boundary conditions, as a advantage over the popular semi-Lagrangian schemes. A simple high order positivity-preserving (PP)-limiter was developed and analyzed to ensure the desirable PP property for numerical solutions. The algorithm was extended to multi-dimensional problems and the VP system via dimensional splitting. A collection of numerical tests demonstrated good performance of the proposed scheme in resolving

complex solutions' structures. Future work consists of extending the scheme to high-dimensional VP simulations.

A Formulation of WENO integration and extrapolation

When approximating J_i^L with third order accuracy, the big stencil is $S(i) = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$ and the small stencils are $S_1(i) = \{x_{i-2}, x_{i-1}, x_i\}$ and $S_0(i) = \{x_{i-1}, x_i, x_{i+1}\}$. We denote $\nu = \alpha\Delta x$, then we have

$$\begin{aligned} c_{i-1}^{(0)} &= \frac{2 + \nu - (2 + 3\nu + 2\nu^2)e^{-\nu}}{2\nu^2}, \\ c_i^{(0)} &= -\frac{2 - \nu^2 - (2 + 2\nu)e^{-\nu}}{\nu^2}, \\ c_i^{(0)} &= \frac{2 - \nu - (2 + \nu)e^{-\nu}}{2\nu^2}, \\ c_{i-2}^{(1)} &= \frac{2 - \nu - (2 + \nu)e^{-\nu}}{2\nu^2}, \\ c_{i-1}^{(1)} &= -\frac{2 - 2\nu - (2 - \nu^2)e^{-\nu}}{\nu^2}, \\ c_i^{(1)} &= \frac{2 - 3\nu + 2\nu^2 - (2 - \nu)e^{-\nu}}{2\nu^2}. \end{aligned}$$

And the linear weights are

$$\begin{aligned} d_0 &= \frac{-(6 - 6\nu + 2\nu^2) - (6 - \nu^2)e^{-\nu}}{3\nu(2 - \nu - (2 + \nu)e^{-\nu})}, \\ d_1 &= \frac{6 - \nu^2 - (6 + 6\nu + 2\nu^2)e^{-\nu}}{3\nu(2 - \nu - (2 + \nu)e^{-\nu})}. \end{aligned}$$

In this case, the smoothness indicators are

$$\begin{aligned} \beta_0 &= \frac{13}{12}(v_{i-1} - 2v_i + v_{i+1})^2 + (v_{i-1} - v_i)^2, \\ \beta_1 &= \frac{13}{12}(v_{i-2} - 2v_{i-1} + v_i)^2 + (v_{i-1} - v_i)^2. \end{aligned}$$

When approximating J_i^L with fifth order accuracy, we choose the large stencil as $S(i) = \{x_{i-3}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, and the three small stencils are $S_0(i) = \{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, $S_1(i) = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$, and $S_2(i) = \{x_{i-3}, x_{i-2}, x_{i-1}, x_i\}$. Then we can obtain

$$\begin{aligned} c_{i-1}^{(0)} &= \frac{6 + 6\nu + 2\nu^2 - (6 + 12\nu + 11\nu^2 + 6\nu^3)e^{-\nu}}{6\nu^3}, \\ c_i^{(0)} &= -\frac{6 + 4\nu - \nu^2 - 2\nu^3 - (6 + 10\nu + 6\nu^2)e^{-\nu}}{2\nu^3}, \end{aligned}$$

$$\begin{aligned}
c_{i+1}^{(0)} &= \frac{6 + 2\nu - 2\nu^2 - (6 + 8\nu + 3\nu^2)e^{-\nu}}{2\nu^3}, \\
c_{i+2}^{(0)} &= -\frac{6 - \nu^2 - (6 + 6\nu + 2\nu^2)e^{-\nu}}{6\nu^3}u_{i+2}, \\
c_{i-2}^{(1)} &= \frac{6 - \nu^2 - (6 + 6\nu + 2\nu^2)e^{-\nu}}{6\nu^3}, \\
c_{i-1}^{(1)} &= -\frac{6 - 2\nu - 2\nu^2 - (6 + 4\nu - \nu^2 - 2\nu^3)e^{-\nu}}{2\nu^3}, \\
c_i^{(1)} &= \frac{6 - 4\nu - \nu^2 + 2\nu^3 - (6 + 2\nu - 2\nu^2)e^{-\nu}}{2\nu^3}, \\
c_{i+1}^{(1)} &= -\frac{6 - 6\nu + 2\nu^2 - (6 - \nu^2)e^{-\nu}}{6\nu^3}, \\
c_{i-3}^{(2)} &= \frac{6 - 6\nu + 2\nu^2 - (6 - \nu^2)e^{-\nu}}{6\nu^3}, \\
c_{i-2}^{(2)} &= -\frac{6 - 8\nu + 3\nu^2 - (6 - 2\nu - 2\nu^2)e^{-\nu}}{2\nu^3}, \\
c_{i-1}^{(2)} &= \frac{6 - 10\nu + 6\nu^2 - (6 - 4\nu - \nu^2 + 2\nu^3)e^{-\nu}}{2\nu^3}, \\
c_i^{(2)} &= -\frac{6 - 12\nu + 11\nu^2 - 6\nu^3 - (6 - 6\nu + 2\nu^2)e^{-\nu}}{6\nu^3}.
\end{aligned}$$

And the linear weights are

$$\begin{aligned}
d_0 &= \frac{60 - 60\nu + 15\nu^2 + 5\nu^3 - 3\nu^4 - (60 - 15\nu^2 + 2\nu^4)e^{-\nu}}{10\nu^2(6 - \nu^2 - (6 + 6\nu + 2\nu^2)e^{-\nu})} \\
d_2 &= \frac{6 - \nu^2 - (6 + 6\nu + 2\nu^2)e^{-\nu}}{3\nu((2 - \nu) - (2 + \nu)e^{-\nu})} \\
d_1 &= 1 - d_0 - d_2
\end{aligned}$$

The smoothness indicators are

$$\begin{aligned}
\beta_0 &= \frac{781}{120}(-v_{i-1} + 3v_i - 3v_{i+1} + v_{i+2})^2 + \frac{13}{48}(-3v_{i-1} + 7v_i - 5v_{i+1} - v_{i+2})^2 + (v_{i-1} - v_i)^2 \\
\beta_1 &= \frac{781}{120}(-v_{i-2} + 3v_{i-1} - 3v_i + v_{i+1})^2 + \frac{13}{48}(v_{i-2} - v_{i-1} - v_i + v_{i+1})^2 + (v_{i-1} - v_i)^2 \\
\beta_2 &= \frac{781}{120}(-v_{i-3} + 3v_{i-2} - 3v_{i-1} + v_i)^2 + \frac{13}{48}(v_{i-3} - 5v_{i-2} + 7v_{i-1} - 3v_i)^2 + (v_{i-1} - v_i)^2
\end{aligned}$$

When doing WENO extrapolation on left boundary, the smoothness indicators are listed below:

$$\begin{aligned}
\beta_1 &= \Delta x^2 \\
\beta_2 &= (v_0 - v_1)^2 \\
\beta_3 &= \frac{13}{12}(v_0 - 2v_1 + v_2)^2 + (2v_0 - 3v_1 + v_2)^2 \\
\beta_4 &= \frac{781}{720}(v_0 - 3v_1 + 3v_2 - v_3)^2 + \frac{13}{48}(5v_0 - 13v_1 + 11v_2 - 3v_3)^2 + (3v_0 - 6v_1 + 4v_2 - v_3)^2
\end{aligned}$$

$$\begin{aligned}\beta_5 = & \frac{1421461}{1310400}(v_0 - 4v_1 + 6v_2 - 4v_3 + v_4)^2 + \frac{781}{720}(-3v_0 + 11v_1 - 15v_2 + 9v_3 - 2v_4)^2 \\ & + \frac{13}{7300800}(3379v_0 - 10786v_1 + 12864v_2 - 6886v_3 + 1429v_4)^2 \\ & + (-4v_0 + 10v_1 - 10v_2 + 5v_3 - v_4)^2\end{aligned}$$

For J_i^R and extrapolation on the right boundary, we can have the coefficients due to the symmetry.

B Formulation of RK(2,3) and RK(4,4)

For $RK(2, 3)$, the matrixes \mathbf{A} and \mathbf{b} are given as

$$\mathbf{A} = \begin{pmatrix} \frac{1}{2}(1 - \frac{1}{\sqrt{3}}) & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{2}(1 - \frac{1}{\sqrt{3}}) \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

When $c > 0$, $RK(2, 3)$ can be rewritten as

$$\begin{aligned}u^{(1)}(x) &= I^L [u^n, \alpha] (x) + A^{rk,1} e^{-\alpha(x-a)}, \\ u^{(2)}(x) &= I^L [-\sqrt{3}u^n + (1 + \sqrt{3})u^{(1)}, \alpha] (x) + A^{rk,2} e^{-\alpha(x-a)} \\ u^{n+1}(x) &= (1 + \sqrt{3})u^n(x) - \frac{3}{2}(1 + \sqrt{3})u^{(1)}(x) + \frac{1}{2}(3 + \sqrt{3})u^{(2)}(x).\end{aligned}$$

where $\alpha = \frac{3+\sqrt{3}}{c\Delta t}$. And the corresponding time on each stage is $t_{rk}^1 = t_n + \frac{1}{2}(1 - \frac{1}{\sqrt{3}})\Delta t$ and $t_{rk}^2 = t_n + \frac{1}{2}(1 + \frac{1}{\sqrt{3}})\Delta t$.

For $RK(4, 4)$, the matrixes \mathbf{A} and \mathbf{b} are given as

$$\mathbf{A} = \begin{pmatrix} 0.087475824368378 & 0 & 0 & 0 \\ 0.306653000581791 & 0.106634669130071 & 0 & 0 \\ 0.306653000581791 & 0.325811845343484 & 0.106634688637712 & 0 \\ 0.306049667930486 & 0.220166571892301 & 0.220166585074543 & 0.087475807723977 \end{pmatrix}$$

$$\mathbf{b}^T = (0.306092539007907 \quad 0.204522170534763 \quad 0.204522182780312 \quad 0.284863107677018)$$

When $c > 0$, we can write $RK(4, 4)$ as

$$\begin{aligned}u^{(1)}(x) &= I^L [u^n, \alpha_1] (x) + A^{rk,1} e^{-\alpha_1(x-a)}, \\ u^{(2)}(x) &= I^L [-2.50557428633555u^n + 3.5055742863355546u^{(1)}, \alpha_2] + A^{rk,2} e^{-\alpha_2(x-a)} \\ u^{(3)}(x) &= I^L [5.149963903410283u^n - 7.205366503722802u^{(1)} + 3.0554026003125183u^{(2)}, \alpha_3] \\ &\quad + A^{rk,3} e^{-\alpha_3(x-a)}\end{aligned}$$

$$\begin{aligned}
u^{(4)}(x) = & I^L[-7.958496756460987u^n + 11.1375658967502u^{(1)} - 4.243749854297317u^{(2)} \\
& + 2.0646807140081034u^{(3)}, \alpha_4] + A^{rk,4}e^{-\alpha_4(x-a)} \\
u^{n+1} = & 18.345647630787816u^n - 25.673987908728908u^{(1)} + 9.877479922993285u^{(2)} \\
& - 4.805618380017192u^{(3)} + 3.256478734964997u^{(4)}
\end{aligned}$$

where

$$\begin{aligned}
\alpha_1 &= 1/(0.087475824368378\Delta t), \\
\alpha_2 &= 1/(0.106634669130071c\Delta t), \\
\alpha_3 &= 1/(0.106634688637712c\Delta t), \\
\alpha_4 &= 1/(0.087475807723977c\Delta t).
\end{aligned}$$

The corresponding time on each stage is

$$\begin{aligned}
t_{rk}^1 &= t_n + 0.087475824368378\Delta t, \\
t_{rk}^2 &= t_n + 0.413287669711862\Delta t, \\
t_{rk}^3 &= t_n + 0.739099534562987\Delta t, \\
t_{rk}^4 &= t_n + 0.833858632621307\Delta t.
\end{aligned}$$

In the case $c < 0$, we have the similar form.

References

- [1] I. Alonso-Mallo. Runge-kutta methods without order reduction for linear initial boundary value problems. *Numerische Mathematik*, 91(4):577–603.
- [2] M. Begue, A. Ghizzo, and P. Bertrand. Two-dimensional Vlasov simulation of Raman scattering and plasma beatwave acceleration on parallel computers. *J. Comput. Phys.*, 151(2):458–478, 1999.
- [3] N. Besse, J. Segré, and E. Sonnendrücker. Semi-lagrangian schemes for the two-dimensional vlasov-poisson system on unstructured meshes. *Transport Theory and Statistical Physics*, 34(3-5):311–332, 2005.

- [4] C. K. Birdsall and A. B. Langdon. *Plasma Physics Via Computer Simulation*. CRC Press, 2005.
- [5] M. Causley and A. Christlieb. Higher order A-stable schemes for the wave equation using a successive convolution approach. *SIAM Journal on Numerical Analysis*, 52(1):220–235, 2014.
- [6] M. Causley, A. Christlieb, B. Ong, and L. Van Groningen. Method of lines transpose: An implicit solution to the wave equation. *Mathematics of Computation*, 83(290):2763–2786, 2014.
- [7] M. F. Causley, H. Cho, A. J. Christlieb, and D. C. Seal. Method of lines transpose: High order l-stable $\mathcal{O}(n)$ schemes for parabolic equations using successive convolution. *SIAM Journal on Numerical Analysis*, 54(3):1635–1652, 2016.
- [8] M. F. Causley, A. Christlieb, Y. Güçlü, and E. Wolf. Method of lines transpose: A fast implicit wave propagator. *arXiv preprint arXiv:1306.6902*, 2013.
- [9] C. Cheng and G. Knorr. The integration of the Vlasov equation in configuration space. *J. Comput. Phys.*, 22(3):330–351, 1976.
- [10] Y. Cheng, A. J. Christlieb, W. Guo, and B. Ong. An asymptotic preserving Maxwell solver resulting in the Darwin limit of electrodynamics. *submitted*, 2015.
- [11] Y. Cheng, A. J. Christlieb, and X. Zhong. Numerical study of the two-species Vlasov-Ampère system: energy-conserving schemes and the current-driven ion-acoustic instability. 2014. submitted.
- [12] Y. Cheng, I. M. Gamba, and P. J. Morrison. Study of conservation and recurrence of runge–kutta discontinuous Galerkin schemes for vlasov–poisson systems. *J. Sci. Comput.*, pages 1–31, 2012.
- [13] C.-S. Chou and C.-W. Shu. High order residual distribution conservative finite difference weno schemes for steady state problems on non-smooth meshes. *J. Comput. Phys.*, 214(2):698–724, 2006.
- [14] C.-S. Chou and C.-W. Shu. High order residual distribution conservative finite difference weno schemes for convection–diffusion steady state problems on non-smooth meshes. *J. Comput. Phys.*, 224(2):992–1020, 2007.

- [15] A. Christlieb, W. Guo, M. Morton, and J.-M. Qiu. A high order time splitting method based on integral deferred correction for semi-Lagrangian Vlasov simulations. *J. Comput. Phys.*, 267:7–27, 2014.
- [16] N. Crouseilles, M. Mehrenberger, and E. Sonnendrücker. Conservative semi-lagrangian schemes for vlasov equations. *Journal of Computational Physics*, 229(6):1927–1953, 2010.
- [17] N. Crouseilles, T. Respaud, and E. Sonnendrücker. A forward semi-lagrangian method for the numerical solution of the vlasov equation. *Computer Physics Communications*, 180(10):1730–1745, 2009.
- [18] F. Filbet and E. Sonnendrücker. Comparison of Eulerian Vlasov solvers. *Comput. Phys. Commun.*, 150(3):247–266, 2003.
- [19] F. Filbet, E. Sonnendrücker, and P. Bertrand. Conservative numerical schemes for the Vlasov equation. *J. Comput. Phys.*, 172(1):166–187, 2001.
- [20] E. Forest and R. Ruth. Fourth-order symplectic integration. *Physica D: Nonlinear Phenomena*, 43(1):105–117, 1990.
- [21] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.
- [22] W. Guo and J.-M. Qiu. Hybrid semi-Lagrangian finite element-finite difference methods for the Vlasov equation. *J. Comput. Phys.*, 234:108–132, 2013.
- [23] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. CRC Press, 2010.
- [24] G.-s. Jiang, C.-W. Shu, et al. Efficient implementation of weighted eno schemes. In *J. Comput. Phys.*, 1996.
- [25] D. I. Ketcheson, C. B. Macdonald, and S. Gottlieb. Optimal implicit strong stability preserving runge–kutta methods. *Applied Numerical Mathematics*, 59(2):373–392, 2009.
- [26] C. Liang and Z. Xu. Parametrized maximum principle preserving flux limiters for high order schemes solving multi-dimensional scalar hyperbolic conservation laws. *J. Sci. Comput.*, 58(1):41–60, 2014.

- [27] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, 1994.
- [28] Y.-y. Liu, C.-w. Shu, and M.-p. Zhang. On the positivity of linear weights in weno approximations. *Acta Mathematicae Applicatae Sinica, English Series*, 25(3):503–538, 2009.
- [29] G. J. Parker and W. N. G. Hitchon. Convected scheme simulations of the electron distribution function in a positive column plasma. *Japanese journal of applied physics*, 36(7S):4799, 1997.
- [30] J. Qiu and C. Shu. Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: Theoretical analysis and application to the Vlasov-Poisson system. *J. Comput. Phys.*, 230(23):8386–8409, 2011.
- [31] J.-M. Qiu and A. Christlieb. A Conservative high order semi-Lagrangian WENO method for the Vlasov equation. *J. Comput. Phys.*, 229:1130–1149, 2010.
- [32] J.-M. Qiu and C.-W. Shu. Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation. *Commun. Comput. Phys.*, 10(4):979–1000, 2011.
- [33] J.-M. Qiu and C.-W. Shu. Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: Theoretical analysis and application to the Vlasov-Poisson system. *J. Comput. Phys.*, 230(23):8386 – 8409, 2011.
- [34] J. Rossmannith and D. Seal. A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations. *J. Comput. Phys.*, 230(16):6203–6232, 2011.
- [35] A. J. Salazar, M. Raydan, and A. Campo. Theoretical analysis of the Exponential Transversal Method of Lines for the diffusion equation. *Numerical Methods for Partial Differential Equations*, 16(1):30–41, 2000.
- [36] M. Schemann and F. Bornemann. An adaptive Rothe method for the wave equation. *Computing and Visualization in Science*, 1(3):137–144, 1998.
- [37] D. C. Seal. *Discontinuous Galerkin methods for Vlasov models of plasma*. PhD thesis, UNIVERSITY OF WISCONSIN–MADISON, 2012.

- [38] C.-W. Shu. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM review*, 51(1):82–126, 2009.
- [39] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo. The semi-lagrangian method for the numerical resolution of the vlasov equation. *Journal of computational physics*, 149(2):201–220, 1999.
- [40] S. Tan, C. Wang, C.-W. Shu, and J. Ning. Efficient implementation of high order inverse lax–wendroff boundary treatment for conservation laws. *Journal of Computational Physics*, 231(6):2510–2527, 2012.
- [41] J. P. Verboncoeur. Particle simulation of plasmas: review and advances. *Plasma Phys. Contr. F.*, 47(5A):A231–A260, 2005.
- [42] T. Xiong, J.-M. Qiu, and Z. Xu. A parametrized maximum principle preserving flux limiter for finite difference rk-weno schemes with applications in incompressible flows. *Journal of Computational Physics*, 252:310–331, 2013.
- [43] T. Xiong, J.-M. Qiu, and Z. Xu. Parametrized positivity preserving flux limiters for the high order finite difference weno scheme solving compressible euler equations. *arXiv preprint arXiv:1403.0594*, 2014.
- [44] T. Xiong, J.-M. Qiu, Z. Xu, and A. Christlieb. High order maximum principle preserving semi-lagrangian finite difference weno schemes for the vlasov equation. *Journal of Computational Physics*, 273:618–639, 2014.
- [45] Z. Xu. Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem. *Math. Comput.*, 83(289):2213–2238, 2014.
- [46] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, 2010.
- [47] X. Zhang and C.-W. Shu. On positivity-preserving high order discontinuous galerkin schemes for compressible euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.

- [48] X. Zhang and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20110153. The Royal Society, 2011.
- [49] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *J. Sci. Comput.*
- [50] T. Zhou, Y. Guo, and C.-W. Shu. Numerical study on Landau damping. *Phys. D*, 157(4):322–333, 2001.